Autonomous Aircraft
A nonlinear approach

Group 1034
IAS10 - 2005 - Aalborg University
by
Finn Jensen
Daniel René Hagen Pedersen

Faculty of Engineering and Science
Aalborg University

Institute of Electronic Systems - Department of Control Engineering

TITLE:

Autonomous Aircraft: A nonlinear approach.

PROJECT PERIOD:

9-10th. semester
September 1st - June 2nd 2005

PROJECT GROUP:

Group 1034 - 2004-2005

GROUP MEMBERS:

Finn Jensen
Danial René Hagen Pedersen

SUPERVISOR:

Jan Helbo
Anders la Cour-Harbo
Department of Control Engineering

PRINTS : 11

Number of pages in report : 83

Number pages in appendix : 30

Total number of pages : 122

ABSTRACT:

A Cessna Skylane 182 model aircraft must be stabilized and flying autonomous.
An aerodynamic model is derived using Computer Fluid Dynamics and wind tunnel data from a real Cessna Skylane 182. Blade Element Theory is used to model the propeller thrust. And the dynamics and kinematics are modeled as a rigid body with 6 Degree of Freedom, using quaternions to describe the orientation of the aircraft.
Nonlinear noninteractive feedback linearization is used to linearize the model. And the aircraft is stabilized using a state space controller, and the orientation is controlled by a quaternion control law. The stability of the system is verified using Lyapunov stability theorem.
The models are implemented in a 3D flight simulator, and the controllers are able to control this simulator. The controllers are not implemented on the aircraft. Test flights were used to validate the models, and test the hardware platform.
Further improvements can be implementing a kalman filter and make the aircraft fully autonomous and designing a path generator and controller.

Preface

This report serve as documentation for the 10th semester project:

"Autonomous Aircraft: A nonlinear approach"

in Intelligent Autonomous Systems a specialization at Aalborg University, Department of Control Engineering.

References to literature are done like [7, p. 500], which refer to "Nonlinear Systems" by Hassan K. Khalil on page 500. The reference can be found in the Bibliography on page 80.

A CD-Rom is attached to the report. This CD-Rom contains Matlab scripts, Simulink models and flight simulator, a controller for the simulator, results from the aerodynamic analysis, and a data logger. The CD-Rom contains also: data sheets over the hardware, software, and movies of a test flight. This report is also on the CDROM found in a pdf format.

A nomenclature list is placed in the start of the report on page v, and contains variables used throughout the report. An index is found on page 112 of the report.

Aalborg University, June 2nd 2005.

Finn Jensen	Daniel R. H. Pedersen

Aerodynamics
–the ultimate artform.

When you get it right
mighty beasts float up into the sky
When you get it wrong
people die.

-Roger Bacon (c1384)

Nomenclature

α	This is the angle of attack and it is defined as the angle between ${}^{\mathbb{B}}\boldsymbol{x}$ and ${}^{\mathbb{B}}v_{\mathbb{B},x}$ in the ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{z}$ plane.
$\bar{q}$	The dynamic pressure $\bar{q} = \frac{\rho V_\infty^2}{2}$.
β	This is the sideslip angle of the aircraft, and it is defined as the angle between ${}^{\mathbb{B}}\boldsymbol{x}$ and ${}^{\mathbb{B}}v_{\mathbb{B},x}$ in the ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{z}$ plane.
ρ	The density of the atmosphere, $\rho = 1.225$ at an altitude of 100m
${}^{\mathbb{B}}\dot{\boldsymbol{\omega}}_{\mathbb{B}}$	Is the angular acceleration of the aircraft and the time derivative of ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$.
${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$	Is the angular velocity of the aircraft with respect to the inertial reference system seen from the body frame $\mathbb{B}$, around the ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$ axes.
${}^{\mathbb{B}}\boldsymbol{\tau}$	is the vector torque around the ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$ axes.
${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}}$	Linear acceleration of the aircraft in ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$ axes.
${}^{\mathbb{B}}\boldsymbol{F}$	Is a force vector along the ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$ axes.
${}^{\mathbb{B}}\boldsymbol{v}_{\mathbb{B}}$	Linear velocity vector of the aircraft in ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$ axes.
V_∞	Velocity of the aircraft relative to the wind.
$\boldsymbol{\delta}$	The control input vector $\boldsymbol{\delta} = \begin{bmatrix}\delta_f & \delta_a & \delta_e & \delta_r\end{bmatrix}^{\mathrm{T}}$
b	The reference wing span
c	The reference chord line length
$C_{{}^{\mathbb{B}}\boldsymbol{\tau}_A}$	The dimension less aerodynamic torque coefficient.
$C_{{}^{\mathbb{B}}\boldsymbol{F}_A}$	The dimension less aerodynamic force coefficient.
S	The reference wing area
$\mathbb{A}$	Aerodynamic coordinate system, with origin in Center of Aerodynamics CoA, with the three axes: ${}^{\mathbb{A}}\boldsymbol{x}$, ${}^{\mathbb{A}}\boldsymbol{y}$, and ${}^{\mathbb{A}}\boldsymbol{z}$.
$\mathbb{B}$	Aircraft body fixed coordinate system with origin in the CoG and the three coresponding axes ${}^{\mathbb{B}}\boldsymbol{x}$,${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$.
$\mathbb{E}$	Earth Centered, Earth Fixed coordinate system (ECEF) with components ${}^{\mathbb{E}}\boldsymbol{x}$,${}^{\mathbb{E}}\boldsymbol{y}$ and ${}^{\mathbb{E}}\boldsymbol{z}$.
$\mathbb{N}$	Navigation coordinate system establishes a reference point for navigating the aircraft, the three axes are denoted as ${}^{\mathbb{N}}\boldsymbol{x}$,${}^{\mathbb{N}}\boldsymbol{y}$ and ${}^{\mathbb{N}}\boldsymbol{z}$.
$\mathbb{R}$	Aircraft fixed reference coordinate system with origin in the CoG and the three coresponding axes ${}^{\mathbb{R}}\boldsymbol{x}$,${}^{\mathbb{R}}\boldsymbol{y}$ and ${}^{\mathbb{R}}\boldsymbol{z}$, always oriented as the Navigation coordinate system $\mathbb{N}$.
Pitch (θ_P)	Is the angle between ${}^{\mathbb{B}}\boldsymbol{x}$ and ${}^{\mathbb{R}}\boldsymbol{x}$ positive counter clockwise around the ${}^{\mathbb{B}}\boldsymbol{y}$ axis.
Roll (θ_R)	Is the angle between ${}^{\mathbb{B}}\boldsymbol{z}$ and ${}^{\mathbb{R}}\boldsymbol{z}$ positive counter clockwise around the ${}^{\mathbb{B}}\boldsymbol{x}$ axis.
Yaw (θ_Y)	Is the angle between ${}^{\mathbb{B}}\boldsymbol{x}$ and ${}^{\mathbb{R}}\boldsymbol{x}$ positive counter clockwise around the ${}^{\mathbb{B}}\boldsymbol{z}$ axis.

Contents

1 Introduction **1**

1.1 The Aircraft 2

1.2 The Aircraft Platform 2

1.3 Platform Requirements 4

1.4 Platform Design 5

2 Requirements **7**

2.1 Instrumentation 7

2.2 Modeling 8

2.3 Control 8

3 Methodology **9**

3.1 Report Structure 11

4 Definitions **12**

4.1 Parts on the Aircraft 12

4.2 Coordinate Systems 13

4.3 Basic Aerodynamics 15

4.4 An Aircraft with six Degree of Freedom 19

5 Aerodynamics **21**

5.1 Aerodynamic Analysis 22

5.2 Propeller 43

5.3 Aerodynamic Model 48

6 Aircraft Dynamics and Kinematics **49**

6.1 Dynamical Model 50

6.2 Kinematic Model . . . 52
6.3 Validation . . . 52

7 Control of Aircraft **54**
7.1 Feedback Linearization . . . 55
7.2 Angular Velocity Controller . . . 59
7.3 Orientation Controller . . . 59
7.4 Controller Performance . . . 63

8 Stability **66**
8.1 Interconnected Systems Approach . . . 66
8.2 The Stability Analysis . . . 68
8.3 The Result . . . 71

9 Test Flight **72**
9.1 Test Flights . . . 73

10 Conclusion **77**
10.1 Perspective . . . 78

Bibliography **79**

A Data Logger **81**
A.1 Gyro Logger . . . 83
A.2 GPS Logger . . . 83
A.3 Servo Logger . . . 85
A.4 Servo Board . . . 86
A.5 Guide to Data Logger . . . 89

B Test Appendix **90**

C Quaternions **95**

D Hardware Implementation **99**

Index **109**

Chapter 1

Introduction

This Project is part of an ongoing process since 2000, to design an autonomous aircraft at Aalborg University.

From 2000 to 2004 the project have been based on a Graupner Trainer 60 model aircraft, and a completely linear control, and modelling approach.

Unfortunately the Graupner aircraft crashed during a test flight in the last project, which damaged the aircraft and flight hardware severely.

Previous Projects

When reviewing the previous projects we learned that the primary task of all projects had been to develop and understanding a model for the aircraft. Unfortunately has the used model, due to the crash, never been validated or verified. Thus we could not be sure that using the previous approach would be sufficient and time effective.

Combining the aircraft project history with our satellite project experience, and courses in non-linear control, we decided to use a more theoretical and non-linear approach to the control and modelling part of the project.

The Experiences

The experience from the last project group was that it was necessary to acquire a large aircraft, because one of the theories to why the last aircraft crashed, was that it simply where to small to carry the flight hardware.

Other experiences said that the flight hardware, where difficult to work with, mainly because of time and power issues. But there also seemed to be problems with the accuracy and noise levels of the sensory system in the previous project.

Since the aircraft project had to start over again it was decided to put more effort into the flight hardware design and aircraft platform, such that the following projects have a better platform to build their projects on and calculated that it could save this project some time, if there where a

working and easy to use platform.

1.1 The Aircraft

To solve the weight problem from the last project a bigger aircraft was bought. A Cessna Skylane 182, which is 26% scale model of the real Cessna Skylane. See the front page of this report for a picture.

The aircraft consist of the following basic parts:

- Cessna Skylane 182 airframe
- Webra BOXER 30-2 FT-Glow engine
- Two Webra Speed 91-P5 silencers
- Three bladed Graupner propeller
- Graupner SC Receiver
- Two 10.3Kgcm Hitec HS-5645MG Digital Ultra Torque servos
- Two analog Futaba servos
- Four 4.4Kgcm Hitex HS-5475HB Standard Digital servos.
- 0.5l fuel tank

The total weight without Flight hardware is: 7.5kg. The aircraft is manually controlled using a Graupner mx-22 remote control.

1.2 The Aircraft Platform

The previous group[2] designed the system shown in figure 1.1. The individual parts on figure 1.1 are described below

PC-104 The system was build on the Simulink toolbox xPC-target. A small real time operation system where installed on the PC-104. It was then possible to download programs into the PC-104, and execute them from a desktop PC. The interface between the PC-104 and the desktop PC was a RS-232 connection.

GPS & GSM Modem A GPS of the type Lassen SK II was used to determine the position, and velocity of the aircraft. It was chosen to used dGPS, so the position can be determined more precise, and a GSM modem of the type Siemens M20 was used.

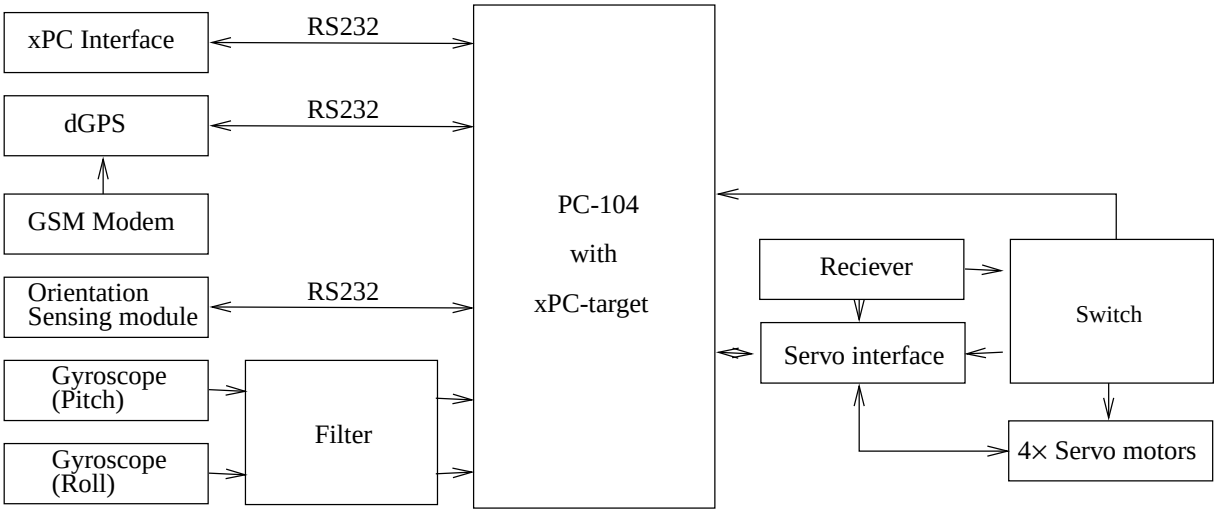


Figure 1.1: *Overview of the system on the Grupner aircraft*

Orientation Sensing module was of the type TCM2 Electronic Compas module. This module measured the pitch and roll angle of the aircraft. The module could also measure the electro magnetic field of the earth in three axis and the temperature of the air. Interface to the PC-104 was a RS232 serial connection.

Gyroscope was used to measure the change of the angle in one direction. Two gyroscopes were used in the aircraft. One measured the angular velocity in pitch direction, and the other gyroscope measured angular velocity in roll direction. The output from the gyroscope was a voltage between -2.5V and 2.5V

Filter The signal from the gyroscopes was filtered to remove noise on the signal.

Switch The aircraft had to be able to fly in 2 modes. One mode was autonomous mode where control algorithms control the aircraft. The other mode was manuel mode where a pilot on the ground controlled the aircraft. The switch was used to shift between the two modes.

Receiver The receiver get the control signals from the pilot on the ground. This receiver used Pulse Code Modulation (PCM). It was a Robbe Futaba PCM 1024.

Servo Motors The actuators, which were used to control the aircraft, were four servo motors. The input to the motors were a PWM signal, which controlled the servomotors.

Servo interface had different tasks in autonomous and manuel mode. In autonomous mode it controlled the servomotors with PWM signals updated with an interval on 20ms. The control signal came from the PC-104. In manuel mode it measured the position of the servos, and sent it to the PC-104. The interface between the PC-104 and the servomotors was implemented in a PIC 16F877.

The experience from the last project [2] showed that XPC eased the implementation. There is two problem which need to be solved, the data storage, and download time, thus it is needed to

find a solution such that the data wasn't lost if the system powered off and such that the download time is shorter. Another reusable design is the switching mechanism between autonomous and manual mode worked.

Another experience from both the last aircraft project and the autonomous helicopter project same year, told that heating and noise was a great issue, thus some effort needed to be placed in the EMC design of the platform, and cooling the platform.

1.3 Platform Requirements

To make this platform usable in future project a feasibility study was conducted, to find the requirements for the hardware platform and the following was the conclusion of this feasibility study.

The conclusion is divided into control tasks for which a sensor type is specified.

Stabilization requires information from the angular rates of the aircraft. The best solution is Gyro measurements of all 3 axes of the aircraft.

Orientation requires measurements to some external reference, here the magnetic field of the earth or the gravitational field could be used.

Flight path The best solution for this is GPS or similar systems, but the accuracy restricts this sensor to pure flight, thus another system is needed during autonomous landing and takeoff.

The feasibility study also contained safety issues and handling issues. This resulted in the following requirements.

Manual override is a requirement to ensure that the aircraft isn't lost if anything should go wrong. Thus it is necessary to ensure that the platform never could take the control of the aircraft without acknowledgement from ground. It should always be possible to manual override any commands from the flight platform.

The power usage of the platform is required to be low, such that the runtime of the hardware platform is extended.

Safe mode It is necessary that the aircraft is manually controllable even if the flight computer is powered off. It is a requirement that the flight computer and servo motors have different power supplies, such that the manual flight time isn't reduced due to errors or short circuits in the flight computer.

Data storage must be non-volatile such that sensor data and logs aren't lost if the flight computer should shutdown.

Download time must be minimized such that it is possible to download data between test flights without waiting to long and using flight power.

Data logging must be carried out in a simple and effective way, such that the measurements of each test case are easy to acquire. The raw data from the measurements should be saved, such that no details are lost in the processing of the data.

1.4 Platform Design

The new platform, which have been designed such that the requirements in section 1.3 are satisfied is described in this section. A user guide for the XPC system can be seen in appendix A.5.

The design is based on two individual parts the critical hardware and Flight Computer (FC) see figure 1.2. The critical hardware covers the Servo Controller and Interface Card (SCIC).

1.4.1 Servo Controller and Interface Card

The SCIC is designed such that it uses a minimum of power and such that it works even if the servo battery voltage drops lower than the input range of the receiver and servos.

- The SCIC is also design such that it are able to measure the servo input signals and this way read the position of the servos.
- Another feature shown on figure 1.2 is the switching mechanism which allows the MCU and Flight computer to control the output signal to the servos. This mechanism is designed such that the ground controller at any time can override the MCU.
- The SCIC is also independent of the FC. Thus no matter which state the FC is in the SCIC always works, as long as there are power on the Servo battery.
- The MCU software is design such that it cannot do any damage to the aircraft, by limiting the servo output ranges.

1.4.2 Flight Computer

As shown on figure 1.2 the FC contains a Power Supply unit (PSU), mass storage device (MASS), GPS, GyroCube, and inteface to: VGA, RS232, reset, keyboard, and mouse.

- The PSU is a switch mode power supply, with a 95% efficiency. It requires an input voltage between 8 – 15V and delivers: 3.3V, and 5.0V output to the entire FC.
- MASS is a 256MB IDE Flash disk, only used for saving measurement data.
- The GyroCube is a combination of a three axes Gyro and a three axes accelerometer. Thus providing the platform with sensors to the stabilize and orientation requirements in section 1.3.

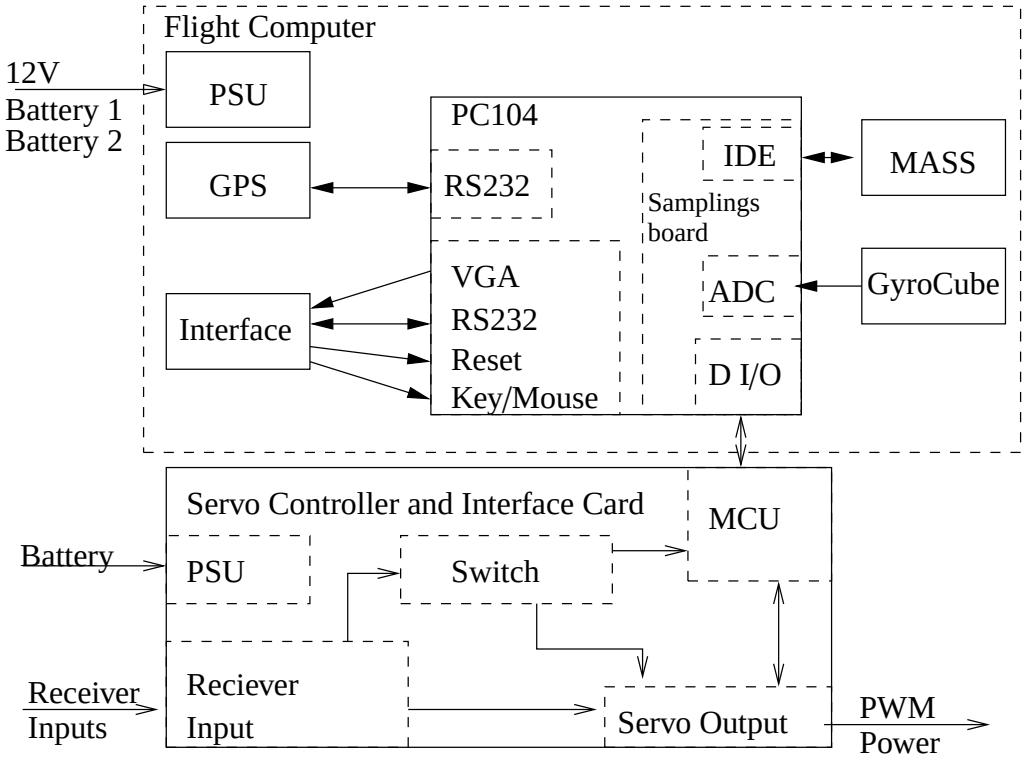


Figure 1.2: *Overview of new hardware platform, including: PC104, Power supply units (PSU), sensors, and Servo Controller and Interface Card (SCIC).*

- The GPS is a SBAS[1] enhanced GPS receiver from Novatel with an accuracy down to 2m[12].

[1]SBAS is ESA's Satellite Based Augmentation System, which provides satellite based differential GPS signal thru EGNOS [4].

Chapter 2

Requirements

This chapter contain the requirements for the project. The requirements are divided in three parts: hardware, software, and instrumentation, a part about modeling the aircraft, and a part about controlling the aircraft.

The main goal for the project is:

The Cessna Skyline 182 model aircraft must be able to fly autonomously.

To reach this goal the following requirement have to be meet. Not all the requirements is equal important, so a prioritizing is made from A to C, where A is the most important.

2.1 Instrumentation

All hardware and software requirements are high priority, because it is necessary for the autonomous flight.

SPEC-1,1 Instrument the Cessna Skylane 182 aircraft model with: sensors, actuators, interfaces, batteries, and flight computer.(**A**)

SPEC-1,2 Implement a data logger including drivers to all sensors and actuators, which are able to log the following:

- Positions (**A**)
- Angular rates (**A**)
- Accelerations (**A**)
- Servo positions (**A**)

SPEC-1,3 Verify that hardware and data logger work in a test flight.(**A**)

SPEC-1,4 In order to prevent the aircraft from crashing these two requirements must be satisfied.

- Implement manual override.(**A**)
- Implement safe mode.(**A**)

2.2 Modeling

SPEC-2,1 Derive a model of the aircraft such that it enables autonomous flight. This includes:

- An dynamic model.(**A**)
- A kinematic model.(**A**)

SPEC-2,2 Validate the model by flying with the aircraft model in a flight simulator.(**A**)

SPEC-2,3 Validate the models with data from test flights.(**B**)

SPEC-2,4 Design a Kalman filter to filter data from the sensor measurements.

- Use it to filter data from the test flight. (**B**)
- Implement a Kalman filter which run under a test flight and is used to filter the measurements for the controller.(**C**)

2.3 Control

SPEC-3,1 Design a nonlinear controller to stabilize the aircraft by controlling the angular velocity.

- Implement the controller on the flight simulator.(**A**)
- Implement the controller on the aircraft.(**B**)

SPEC-3,2 Design a controller to control the orientation of the aircraft.

- Implement the controller on the flight simulator.(**A**)
- Implement the controller on the aircraft.(**C**)

Chapter 3

Methodology

This chapter contains a description of the overall approach of modeling and controlling the aircraft. The different methods used in modeling and control are described in this chapter. This chapter is also used to tell how the rest of the report is structured. The last section of this chapter contains a short description of the chapter contents throughout the report.

The objective of this project is to make a model aircraft fly autonomously. In order to do this it is necessary to make a model of the aircraft, and design a controller, which can control this model. This has been done with the design in figure 3.1. This design split the model part up in four parts, an aerodynamic model, a propeller model, a dynamic model, and a kinematic model. It is chosen not to model the servomotors, because the they have a position controller, which can keep the position. They react also so fast compared with the rest of the system, that the dynamic from the servos would have no influence on.

The controller is split in: a feedback linearization, an angular velocity controller, and an orientation controller. The methods chosen to solve each part is described in the following:

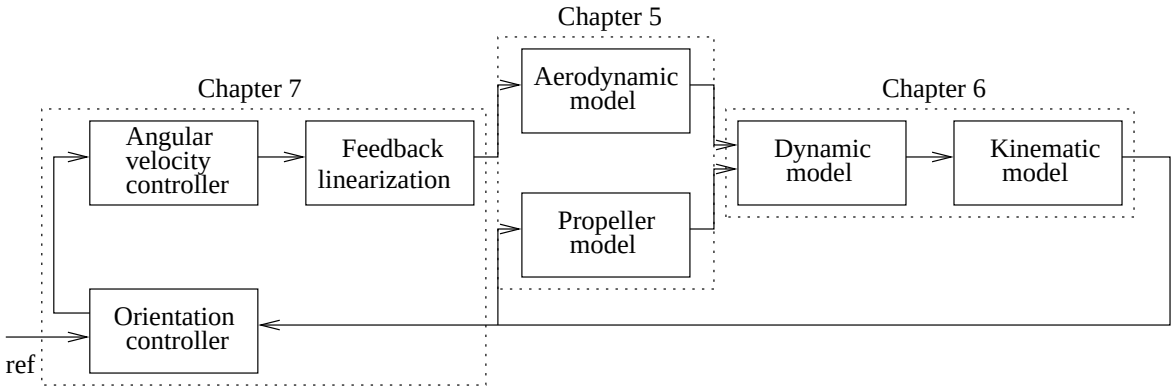


Figure 3.1: *The figure shows the main parts of the system, which include an aerodynamic model, a propeller model, a dynamic model, a kinematic model, feedback linearization, an angular velocity controller, and an orientation controller.*

Aerodynamic model is used to calculate the size of the forces generated on the aircraft, when it is flying through the air. Dimensionless coefficients are used to calculate these forces. This method is chosen because the coefficients is independent of the size of the aircraft, only the shape matters. It is therefore possible to take the coefficients from a real Cessna Skyline 182, and used these coefficients on the model aircraft. Two methods are used to calculate these coefficients. Data from a wind tunnel test for a real Cessna 182 is combined with the Computer Fluid Dynamics (CFD) to calculate the coefficients. The wind tunnel data give the accurate values, while CFD give a better model understanding. The output from this model is coefficients used to calculate the forces and torques acting on the aircraft. The calculation of the coefficients demands a lot of CPU power, and it is impossible to make the calculation in real time on the PC-104. The result is therefore fitted with polynomials, which is execute must faster.

Propeller model calculate the thrust generated by the propeller when the aircraft is flying through the air. It is not possible to control the propeller autonomously, only the pilot would have control over the propeller. The propeller is therefor handled as a disturbance. The model is used to find the size of the thrust generated by propeller as a function of the angular velocity of the propeller and the velocity of the aircraft. The model for the propeller is found by analyzing the generation of thrust in the working area for the propeller. A polynomial is then fitted to the result of the analysis.

Dynamic model calculate velocity and angular velocity of the aircraft, when forces and torques is applied to the system. This model treat the aircraft as a rigid body with 6 degree of freedom. The input for this model is the output from the calculations in the aerodynamic model, and the propeller model.

kinematic model calculate the orientation of the aircraft. The purpose of the project is to make the aircraft fly autonomously where the orientation of the aircraft is controlled, but is should not be able to track a path. It is therefore not necessary to have a kinematic model describing the position of the aircraft. The orientation of the aircraft is described using quaternions. This method is chosen because the solution contains no singularities. A rotation contains only matrix multiplications, which is well suited for a computer.

Feedback linearization linearize the three models with respect to the angular velocity. This method is a nonlinear control technique, which require a precise model, because nonlinearities is canceled out.

Angular velocity controller is a state space control, which stabilize the angular velocity.

Orientation controller is a second control loop, which control the orientation represented by a quaternion.

The main tools used in this project are Matlab and Simulink.

3.1 Report Structure

This section contains a description of the structure for the remainder of the report. Chapter 4, 5, and 6 are modeling of the aircraft. Chapter 7, and 8 are control of the aircraft, while 9 is the result of the test flights.

Chapter 4 contains the definitions used in the entire report. This include defining the important parts of the aircraft, definition of the different coordinate systems, defining basic aerodynamic terms, and explain the notation used in this report.

Chapter 5 contains the aerodynamic analysis of the aircraft. This include finding the aerodynamic coefficients, and derive the aerodynamic model. This chapter include also the propeller model, because it is very similar methods, which is used to find the two models.

Chapter 6 describes the rigid body dynamics and kinematics model of the aircraft.

Chapter 7 contains the controller design, including a nonlinear noninteracting feedback linearization, an angular velocity controller, and a orientation controller.

Chapter 8 contains a stability analysis of the aircraft model, to check the model for stability.

Chapter 9 contains the results of the test flights, which has been performed with the aircraft.

Chapter 4

Definitions

This chapter contains definitions used through out the project. Starting with naming the basic parts of the aircraft. Then defining relevant coordinate systems used to structure and simplify the modeling and control of the aircraft. Followed by a description of parameters and nomenclatures used in the project.

4.1 Parts on the Aircraft

This section defines the parts of the aircraft, which is relevant for modeling and control, see figure 4.1.

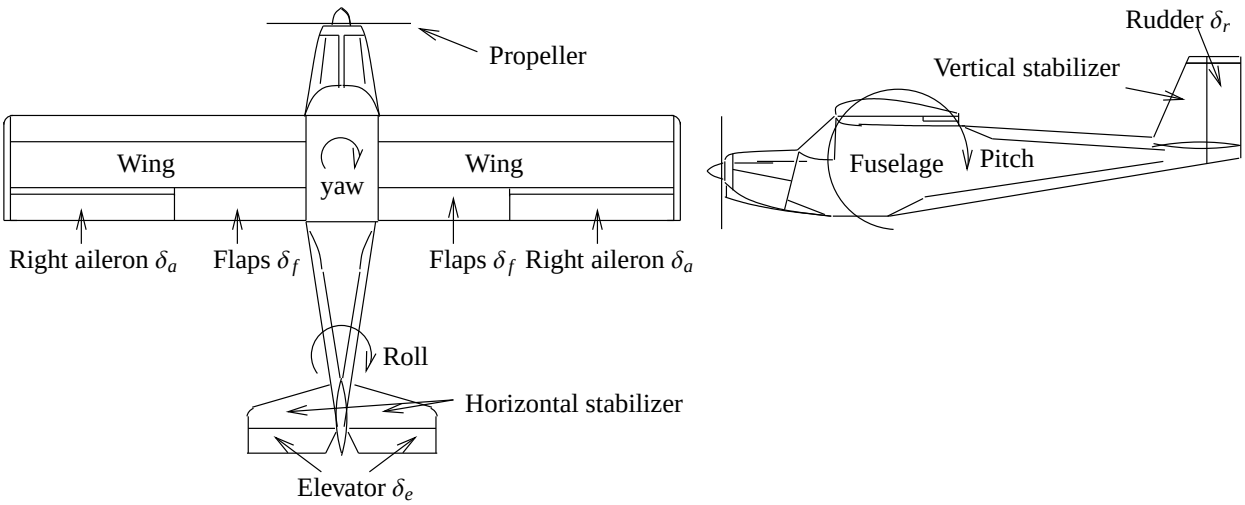


Figure 4.1: *The different parts of the aircraft is defined.*

Wing The function of the wing is to generate the main part of the lift necessary to make the air craft fly. There is two control surfaces on the wing: aileron, and flaps.

Aileron is used to change the roll[1] rate of the aircraft. Since the two ailerons work opposite of each other. The control input to the aileron is the deflection angle δ_a, which is the angle between the two aileron, such that the angle is positive counter clockwise the ${}^{\mathbb{B}}\boldsymbol{y}$-axis.

Flaps are used as air brakes, to reduce the absolute speed of the aircraft during landing. The control input is defined as δ_f

Horizontal stabilizer is used to stabilize the pitch angle.

Elevator is the control surface used to control the pitch rate. The control input is the deflection angle δ_e, which is the angle between the elevator in start position and the current angle. Positive counter clockwise the ${}^{\mathbb{B}}\boldsymbol{y}$ axis.

Vertical stabilizer is used to stabilize the yaw rate.

Rudder is the control surface used to control the yaw rate. The control input for the rudder is the deflection angle δ_r, which is the angle between the rudder in start position and the current position, positive counter clockwise the ${}^{\mathbb{B}}\boldsymbol{z}$ axis.

Fuselage is the body of the aircraft. it contain the flight computer, sensors, and batteries for servomotor and flight computer.

Propeller is the thrust generating part of the aircraft.

4.2 Coordinate Systems

In order to model and control the aircraft it is necessary to define several coordinate systems [13, p. 320]. The different coordinate systems are shown on figure 4.2 and 4.3, and is defined in the following.

Earth-Fixed coordinate system ($\mathbb{E}$) rotates with respect to the earth and is centered in the middle of the earth. It rotates around ${}^{\mathbb{E}}\boldsymbol{z}$, which points toward the north pole. ${}^{\mathbb{E}}\boldsymbol{x}$ goes through the intersection of the Greenwich meridian and equatorial plane, ${}^{\mathbb{E}}\boldsymbol{y}$ is perpendicular to both ${}^{\mathbb{E}}\boldsymbol{x}$ and ${}^{\mathbb{E}}\boldsymbol{z}$ see figure 4.2. This coordinate system is referred to as the Earth-Centered, Earth-Fixed coordinate system abbreviated ECEF. The GPS receiver gives the position in this coordinate system. We use the this coordinate system as in inertial system[2] , which is valid, since the aircraft fly in a small area in short time.

Aircraft body fixed coordinate system ($\mathbb{B}$) is fixed to the aircraft with origin in Center Of Gravity (CoG). Figure 4.3 shows the placement of $\mathbb{B}$ in the aircraft. ${}^{\mathbb{B}}\boldsymbol{x}$ lies on the aircraft centerline through the tip and CoG, positive in the direction of forward motion. ${}^{\mathbb{B}}\boldsymbol{y}$ is perpendicular to ${}^{\mathbb{B}}\boldsymbol{x}$, aligned with the wing and positive in the direction of the right wing tip. ${}^{\mathbb{B}}\boldsymbol{z}$ points downwards and is perpendicular to the two other axes ${}^{\mathbb{B}}\boldsymbol{x}$ and ${}^{\mathbb{B}}\boldsymbol{y}$ see figure 4.3.

[1] Roll, pitch, and yaw is a rotation around ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$, and ${}^{\mathbb{B}}\boldsymbol{z}$.

[2] An Inertial system is coordinate system which is at rest thus haveing either zero or a constant velocity.

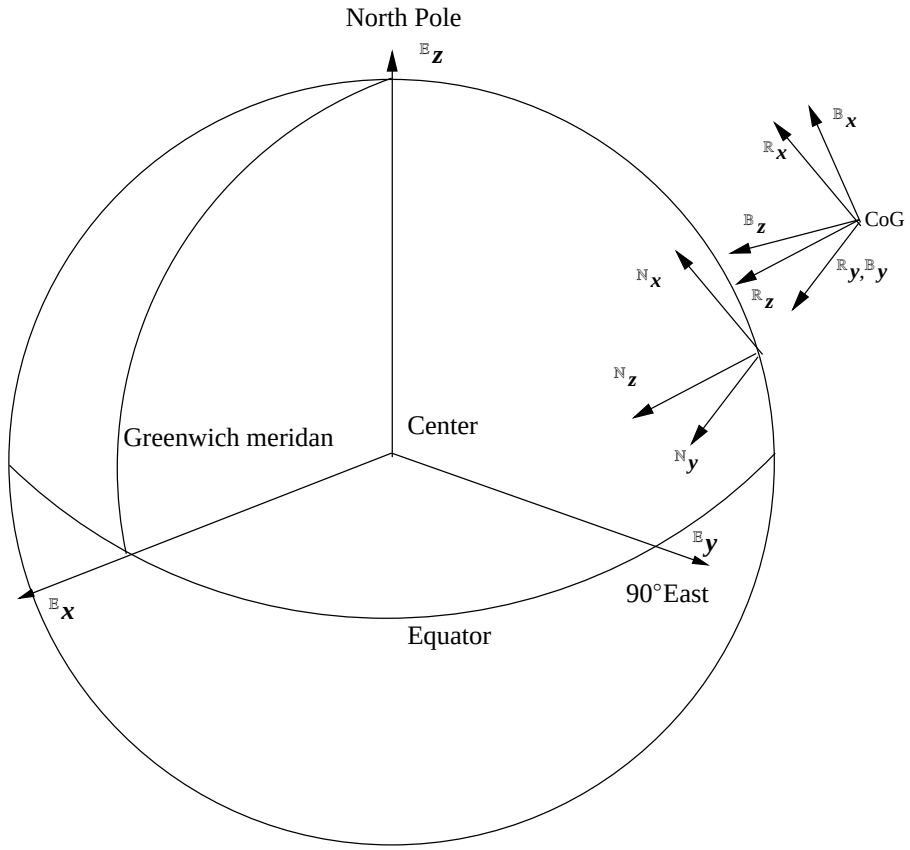


Figure 4.2: *Shows Earth-fixed $\mathbb{E}$, Aircraft body$\mathbb{B}$ and Aircraft Reference $\mathbb{R}$ coordinate systems are related to each other.*

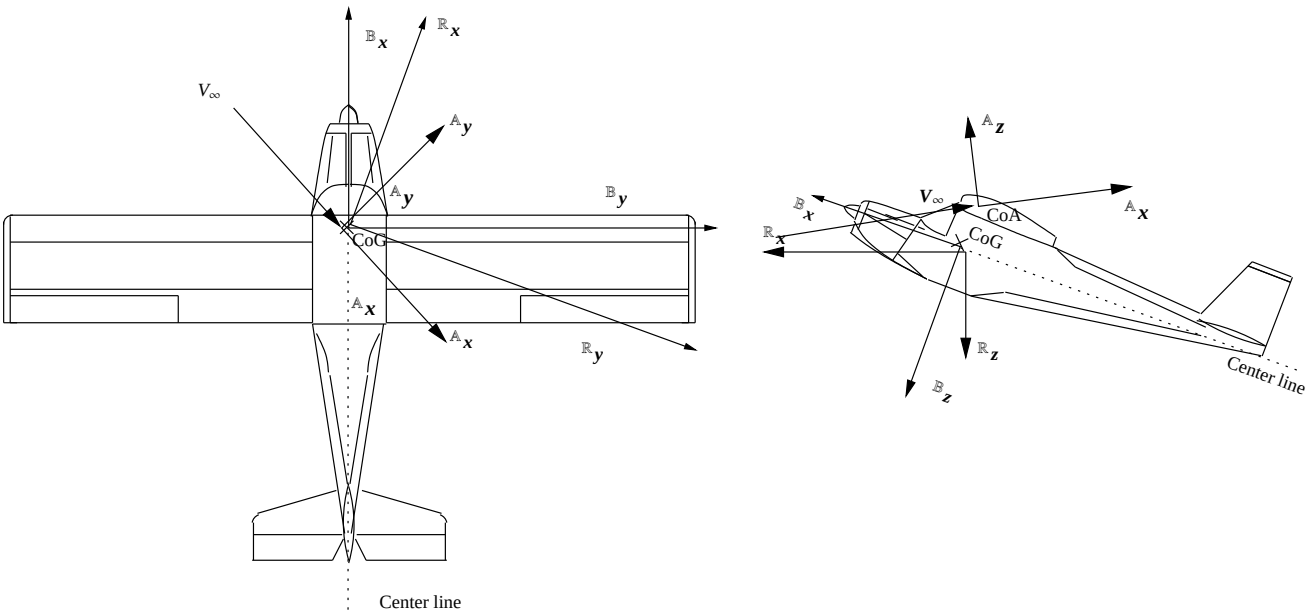


Figure 4.3: *The figure show the placement of the body fixed coordinate system $\mathbb{B}$ and the Aircraft fixed reference coordinate system $\mathbb{R}$ in the orientation for level flight.*

Aircraft fixed reference coordinate system ($\mathbb{R}$) is used to establish a reference for the controller. It is centered in CoG of the Aircraft and the orientation is given by the field where the aircraft takeoff and land. ${}^{\mathbb{R}}\boldsymbol{z}$ points toward the center of the earth. ${}^{\mathbb{R}}\boldsymbol{x}$ is parallel with the runway. ${}^{\mathbb{R}}\boldsymbol{y}$ is perpendicular to the two other axes.

Aerodynamic coordinate system ($\mathbb{A}$) is used in the analysis of: the aerodynamic forces. $\mathbb{A}$ has origin in the Center of Aerodynamics (CoA) approximately $\frac{C}{4}$ from the leading edge[3] of the wing. ${}^{\mathbb{A}}\boldsymbol{x}$ is parallel with ${}^{\mathbb{B}}\boldsymbol{v}_{\mathbb{B}}$, and positive in the direction of $\boldsymbol{V}_{\infty}$. ${}^{\mathbb{A}}\boldsymbol{y}$ and ${}^{\mathbb{A}}\boldsymbol{z}$ is aligned with ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$ when α and β are zero.

Navigational coordinate system ($\mathbb{N}$) is used to describe the position of the aircraft with respect to starting point of the flight. $\mathbb{N}$ is aligned with $\mathbb{R}$, and fixed in the point where the origin of $\mathbb{R}$ is placed in the start of a flight.

4.2.1 Rotation between coordinate systems

Several methods exits to describe the orientation of $\mathbb{B}$ with respect to $\mathbb{E}$ and to rotate a vector from $\mathbb{B}$ to $\mathbb{E}$. In this project rotations are done using quaternions. Appendix C on page 95 describes quaternions and how the are used to rotate a vector from one frame to another. Some properties of Quaternions are that: there are no singularities in the solution, only four parameters is used instead of nine as for Euler angles, and it is less computer demanding.

The nomenclature for a quaternion is in this project as follows: ${}^{\mathbb{B}}_{\mathbb{E}}\boldsymbol{q}$ is the quaternion, which represent the rotation from $\mathbb{E}$ to $\mathbb{B}$.

$$ {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} = {}^{\mathbb{B}}_{\mathbb{E}}\boldsymbol{q} \otimes \begin{bmatrix} {}^{\mathbb{E}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{E}}\boldsymbol{q}^{*} \tag{4.1} $$

Where: ${}^{\mathbb{E}}\boldsymbol{\omega}_{\mathbb{B}}$ is interpreted as the angular velocity of $\mathbb{B}$, measured in $\mathbb{E}$ with respect to $\mathbb{E}$ our inertial system. ${}^{\mathbb{B}}_{\mathbb{E}}\boldsymbol{q}^{*}$ is the complex conjugated quaternion of ${}^{\mathbb{B}}_{\mathbb{E}}\boldsymbol{q}$, $\otimes$ represent quaternion multiplication, and ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ is the angular velocity of $\mathbb{B}$, still measured with respect to $\mathbb{E}$ but seen from or rotated to $\mathbb{B}$. (4.2) rotates back again.

$$ {}^{\mathbb{E}}\boldsymbol{\omega}_{\mathbb{B}} = {}^{\mathbb{B}}_{\mathbb{E}}\boldsymbol{q}^{*} \otimes \begin{bmatrix} {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{E}}\boldsymbol{q} \tag{4.2} $$

4.3 Basic Aerodynamics

This section defines the basic aerodynamic forces: lift, drag, and side-force, which is generated when the aircraft is flying through the air. A model is derived in chapter 5, covering the aerodynamic forces and torques. In order make this model the geometry of the airfoils are defined.

[3] Chord line and leading edge is defined on figure 4.6

4.3.1 Aerodynamic Forces

In general an aircraft is flying because a force called lift is greater than the gravitational force. As seen on figure 4.4 four forces is acting on an aircraft: Lift, Drag, Thrust and Gravity. Gravity is strait forward to determine. The difficult part is: thrust, lift, and drag forces. These forces are determined by the aerodynamics that describes the forces generated by an object moving through gases(atmosphere).

Lift

When a object is moving through a gas a resulting force is created on the object. This force depends of: the geometry of the object moving through the gas, velocity of the object, angle of attack, and slipping. The component of the force which is perpendicular to the direction of motion between the gas and the aircraft is called Lift. If the lifting force gets greater than the gravity force then the aircraft is flying.

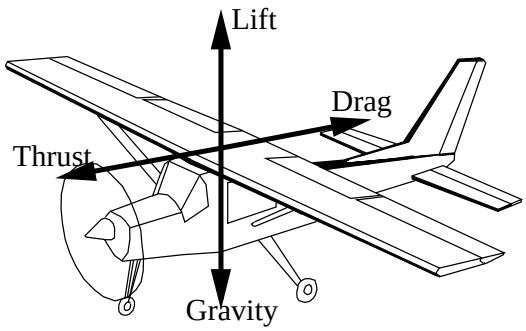


Figure 4.4: *Shows the four forces Lift, Drag, Thrust and gravity on the aircraft*

Drag

In general drag is the force, generated by a solid object moving through a fluid or gas, oppose the direction of movement. There are different sources for drag: one is Skin Friction which is generated due to the friction between molecules and the solid object[3]. Another source is the form drag which is generated due to pressure changes around an object and adding all the forces up, the form drag force is the force projected onto the vector oppose the direction of movement through the air[3], also defined as the ${}^{\mathbb{A}}\boldsymbol{x}$-axis.

Thrust

Thrust is caused by the propeller, which in fact are rotating wings. This rotation courses a lifting force to be created in the direction of ${}^{\mathbb{B}}\boldsymbol{x}$. This lifting force is defined as the Thrust.

Torque

The two forces lift and drag are generated all over the aircraft and vary with the control surfaces: Flaps, Aileron, Elevator, and Rudder. These force are applied at different distances to the CoG thus causing a torque on the aircraft and making it rotate around ${}^{\mathbb{B}}\boldsymbol{x}$,${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$.

4.3.2 Airfoil

The airfoil geometry is an important factor, when lift and drag are calculated. The geometry used to describe an airfoil is defined in this section. A 3 dimensional wing is shown on figure 4.5, while figure 4.6 shows the wing in ${}^{\mathbb{B}}\boldsymbol{x}{}^{\mathbb{B}}\boldsymbol{z}$-plane. The front of the wing is called the

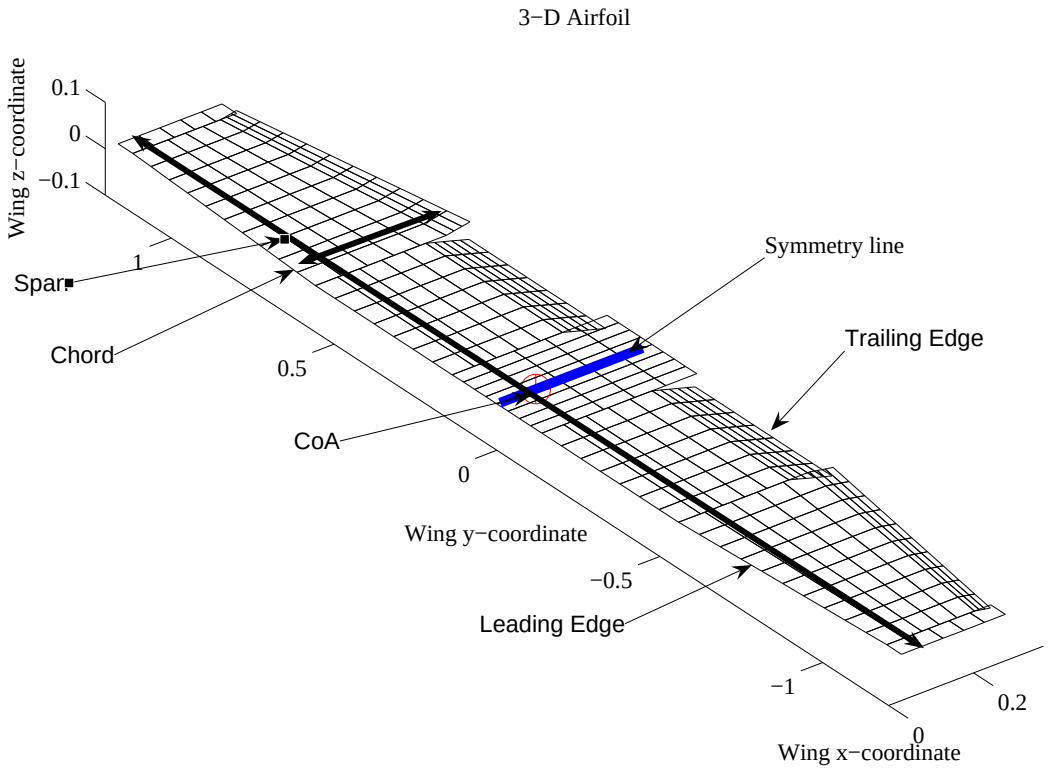


Figure 4.5: *The figure show: leading edge, trailing edge, CoA, Chord, and span, on the wings.*

leading edge, the back of the wing is called the trailing edge. The length between the leading and the trailing edge is the chord length. Because the wing is not a rectangle the chord vary along the span. An important property of the wing is the Center of Aerodynamic pressure

(CoA). CoA is for aircraft with low velocity placed on a axes which lie a $\frac{1}{4}$ chord from the leading edge [3]. CoA is placed on the symmetry line. The aerodynamic forces lift and drag is generated in this point, while the torque work around the axes of $\mathbb{B}$.

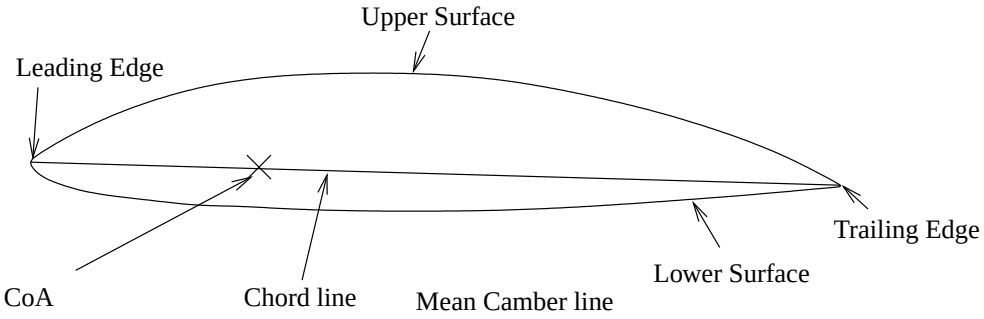


Figure 4.6: *An airfoil is shown with: surfaces, chord line, leading edges, Trailing edge, and CoA.*

4.3.3 Important Aerodynamic Parameters

Important parameters for the generation of lift and drag is: geometry of the wing, angle of attack, and slipping. Slipping and angle of attack is defined in this section.

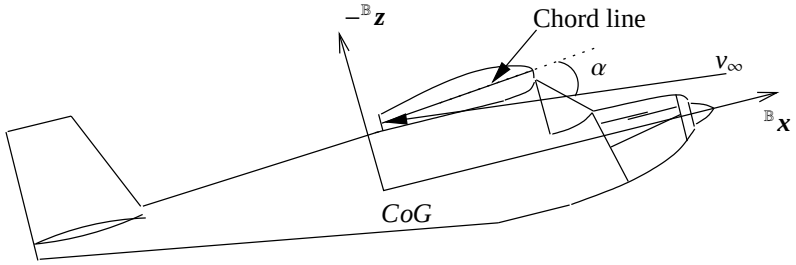


Figure 4.7: *Definition of angle of attack α, and V_∞.*

V infinity is the relative wind speed(V_∞). It is the speed of the air toward the aircraft. It is assumed that there is no wind, thus $V_\infty = |-{}^{\mathbb{B}}\boldsymbol{v}|$.

Angle of attack Angle of attack (α) is an important parameter for the stability of the aircraft in the vertical plane. The lift of the plane is dependent of this parameter, and a change in α gives changes in lift and drag. α is defined as the angle between the chord line in the wings, see section 4.3.2, and velocity vector $\boldsymbol{V}$, thus $\alpha = \arctan\left(\frac{{}^{\mathbb{B}}v_z}{{}^{\mathbb{B}}v_x}\right)$.

Side slipping in the horizontal plane, the slipping angle (β) an important stability parameter and can be seen on figure 4.8. It describes the angle between ${}^{\mathbb{B}}\boldsymbol{x}$ and the velocity vector ${}^{\mathbb{B}}\boldsymbol{v}_{\mathbb{B}}$, in the ${}^{\mathbb{B}}\boldsymbol{x}{}^{\mathbb{B}}\boldsymbol{y}$-plane. β is given as: $\beta = \arcsin\left(\frac{{}^{\mathbb{B}}v_{\mathbb{B},y}}{V_\infty}\right)$

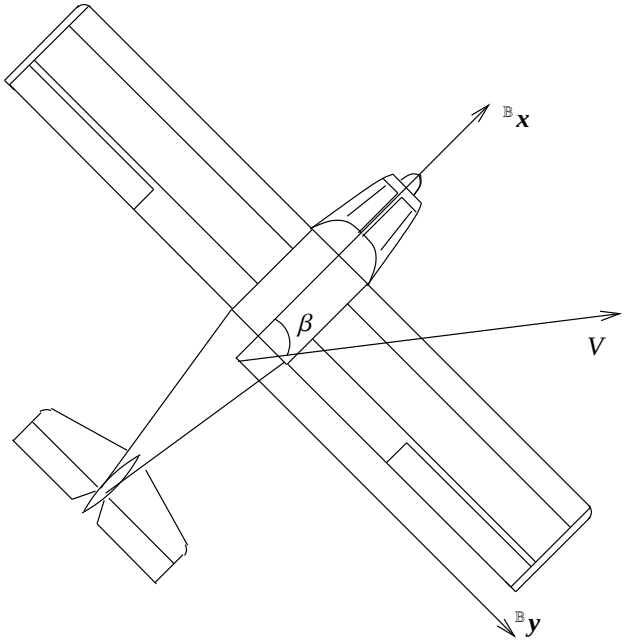


Figure 4.8: *Definition of slipping angle β.*

4.4 An Aircraft with six Degree of Freedom

The aircraft can perform six different motions. Three of these are translatorial motions along the axes of $\mathbb{B}$ with respect to $\mathbb{E}$ and three are angular rotation of $\mathbb{B}$ with respect to $\mathbb{E}$. This gives a system with 6 Degrees of Freedom (DoF), which is shown on figure 4.9. The figure also shows the notation used for the vector components of: force, torques, velocity, and acceleration.

Forces, velocities and accelerations are all given in vectors, which have three components as it is shown in figure 4.9. As an example the angular velocity vector $^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ shown in (4.3). This vector is seen from $\mathbb{B}$ and measured with respect to $\mathbb{E}$.

$$^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} = \begin{bmatrix} ^{\mathbb{B}}\omega_{\mathbb{B},x} \\ ^{\mathbb{B}}\omega_{\mathbb{B},y} \\ ^{\mathbb{B}}\omega_{\mathbb{B},z} \end{bmatrix} \tag{4.3}$$

A description is given of all the vectors used to describe motions and forces in table 4.1.

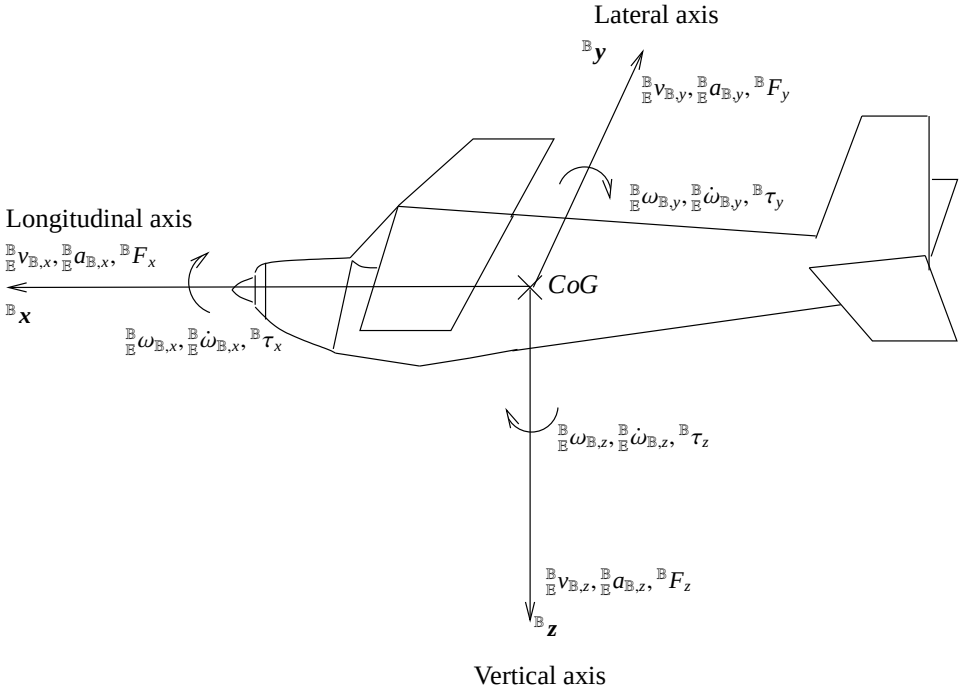


Figure 4.9: *Definition of: velocity, acceleration, forces, angular velocity, torques, and angular accelerations on the aircraft.*

Unit	Description	Type
$\boldsymbol{v}$	Velocity of the aircraft	Translatorial
$\boldsymbol{a}$	Accelerations	Translatorial
$\boldsymbol{F}$	Linear forces	Translatorial
$\boldsymbol{\omega}_{\mathbb{B}}$	Angular velocity of $\mathbb{B}$	
$\dot{\boldsymbol{\omega}}_{\mathbb{B}}$	Angular acceleration of $\mathbb{B}$	Angular
$\boldsymbol{\tau}$	Torque	Angular

Table 4.1: *Forces, velocities, and accelerations on the aircraft.*

Chapter 5

Aerodynamics

This chapter contains an analysis of forces and torques generated by the air flow around the Cessna 182 Skylane aircraft. The flow arround the aircraft is analysed through Computer Fluid Dynamics (CFD) simulations. The CFD simulation results are combined with Wind tunnel data [14] to model the aircraft more accurately. The resulting forces and torques are then determined from the general aerodynamic force and torque equations.

To simplify the modelling process we choose to divide the aircraft aerodynamics into two parts: The aerodynmics, and the Propeller dynamics, as on figure 5.1.

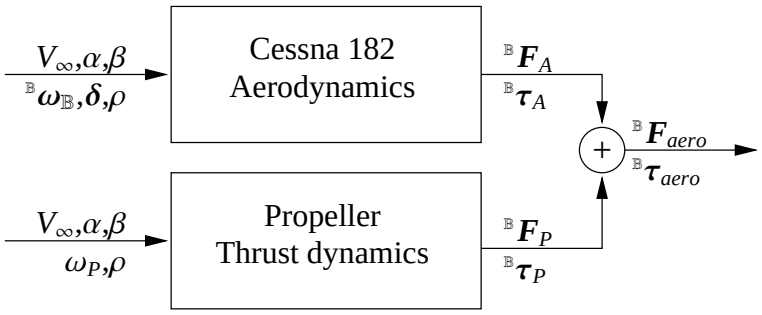


Figure 5.1: *The different aerodynamic models covered in this chapter: Cessna 182 model, and Propeller thrust model.*

In addition figure 5.1 shows the inputs to the models: Density of the atmosphere ρ, Relative wind speed V_∞, angle of attack α, sideslip angle β, control inputs $\boldsymbol{\delta} = \begin{bmatrix}\delta_f & \delta_a & \delta_e & \delta_r\end{bmatrix}^{\mathrm{T}}$, ω_P the angular velocity of the propel, and ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ the angular rates of the aircraft.

These inputs are feed to the aerodynamic models, which returns the respective aerodynamic forces: ${}^{\mathbb{B}}\boldsymbol{F}_A$, and ${}^{\mathbb{B}}\boldsymbol{F}_P$, and torques: ${}^{\mathbb{B}}\boldsymbol{\tau}_A$, and ${}^{\mathbb{B}}\boldsymbol{\tau}_P$. The sum of the forces is denoted as ${}^{\mathbb{B}}\boldsymbol{F}_{aero}$ and the torques ${}^{\mathbb{B}}\boldsymbol{\tau}_{aero}$.

The main problem in modeling aerodynamics is to find an expression for the aerodynamic forces and torques. For this purpose we use the standard equation for aerodynamic force (5.1)

and aerodynamic torque (5.2) [3],

$$^{\mathbb{B}}\boldsymbol{F}_A = \bar{q}S\,C_{^{\mathbb{B}}\boldsymbol{F}_A} \tag{5.1}$$

$$^{\mathbb{B}}\boldsymbol{\tau}_A = \bar{q}S \begin{bmatrix} b & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & b \end{bmatrix} C_{^{\mathbb{B}}\boldsymbol{\tau}_A} \tag{5.2}$$

where $\bar{q}$ is the dynamic pressure and S ,b,and c are the reference: Surface area, span, and chord length . $C_{^{\mathbb{B}}\boldsymbol{F}_A}$ and $C_{^{\mathbb{B}}\boldsymbol{\tau}_A}$ is the dimension less aerodynamic force and torque coefficients. The dimension less property of the coefficients, is used such that the coefficients don't depends on the size and length of the aircraft only the shape. Instead special reference lengths b and c are used to scale the size of the aircraft.

The aerodynamic coefficients characterise the aerodynamic properties of the aircraft. Thus it is essential to find these coefficients to model the aerodynamics. All aerodynamic literature in this project is based on designing an aircraft, which full fill a curtain aerodynamic characteristic. This Project is the reversed problem, the aerodynamics properties of a given aircraft must be found. Thus the methods used in this project are based on a reverse engineered design of an aircraft.

To find the coefficients, there are different design approaches: wind tunnel testing, test flights or CFD simulations. This project uses a combination of all three methods. The reason for using a combination is to reduce the final model error, thus making better conditions for the nonlinear feedback linearization later on in chapter 7.1, because the linearization method requires a precise model of the aircraft, to be able to cancel the right nonlinearities. Since the simulations are very computational demanding we will use polynomial fits to calculate the coefficients at runtime. This approach reduces the computational demand of the model.

5.1 Aerodynamic Analysis

This section contains an aerodynamic analysis of the Cessna 182 Skylane. The analysis of the propeller will follow in section 5.2.

To model the aerodynamic forces on an aircraft, it is nescessary to find the aerodynamic characteristics. This could be done in various ways, where a simple and efficient method is the Vortex Lattice Method (VLM) [8], which is described briefly in the following section. Another more accurate method is to perform experimental wind tunnel tests. In this project we derive our aerodynamic model from both methods, and we use test flights to validate the model.

5.1.1 Vortex Lattice Method

VLM is a method for modelling flow around objects. This is done by panelizing the surface of the airfoils see figure 5.2 and calculating the airflow on each panel. This method is unfortunately very computational demanding, thus not suitable for online calculations. VLM uses

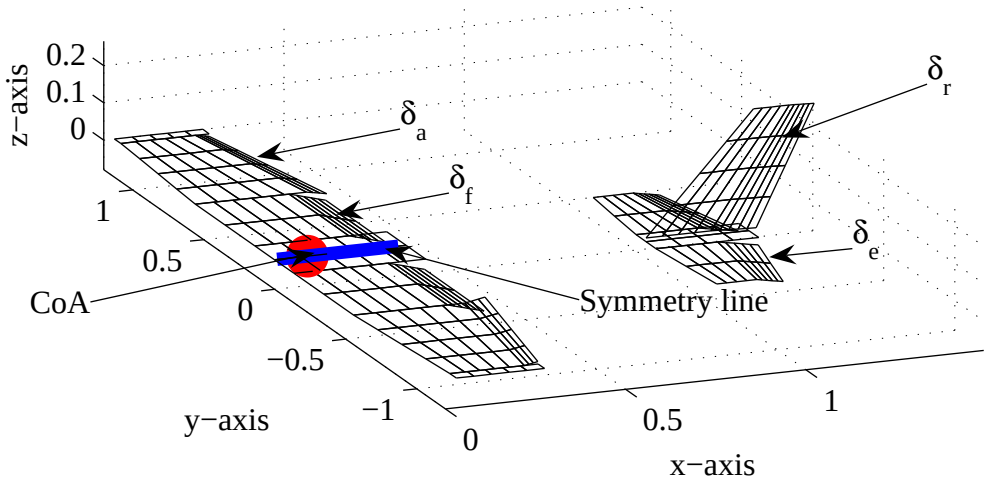


Figure 5.2: *The panelized airfoil configuration of the Cessna 182 Skylane, with $\delta_f = 15°$ flaps, aileron deflection $\delta_a = 10°$, elevator deflection $\delta_e = 10°$, and rudder deflection $\delta_r = -10°$. The airfoils is visualised in the $\mathbb{A}$ coordinate system, with origin at the intersection of leading edge and the symmetry line of the wing. The CoA is shown in the point where the Mean Aerodynamic Chordline (MAC) and Symmetry line intersects marked by a ball.*

the Cartesian coordinate system referred to as the aerodynamic coordinate system $\mathbb{A}$, which is shown on figure 5.2 with $\alpha = 0$, and $\beta = 0$ and described in section 4.2.

VLM is restricted to describing inviscid and incompressible flow, meaning low speed subsonic flow[1], where skin friction is disregarded. These flows are efficient modelled through an idealised flow theory called Potential Flow Theory[8], which is used in the VLM together with thin airfoil theory[8], which disregards thickness effects[2].

In Potential Flow Theory any flow is modelled as a superposition of the elementary flow types: Uniform flow, Sinks and Sources, Doublets, and Vortices see the description below:

Uniform flow is a constant flow, like the flow of water from a water tap.

Sinks and sources is describing the flow, where water is flowing to or from a single point. The drain from a sink is a Sink flow and the flow into a sink is a Source.

Doublets are a special combination of sinks and sources placed in the same point.

Vortices is describing the rotating flows, like when water runs down the drain of sink it makes a spiral turning flow.

[1] A low speed subsonic flow is a flow with a Mach number $M = \frac{V}{a} < 0.2$, a is the speed of sound $340.3^{m}/_{s}$ and V is the flow speed. This is a rule of thumb stating: that most airfoil shapes never experience local supersonic flow at Mach numbers < 0.2.

[2] Thickness effects are the effects caused by the shape of the airfoil, where thin airfoil theory only investigates the aerodynamics of the mean chord line curve.

To describe the flow over an airfoil a combination of all the elementary flow are needed, but experiments have shown that to describe the lifting forces a combination of vortices and uniform flow is sufficient to a high degree of accuracy [8].

VLM is designed to model lifting forces, thus it uses the combination of vortices and the uniform flow to describe the flow around an airfoil. The induced flow[3] from all panels including itself are calculated and summed up, to determine the flow on each panel. A Matlab implementation of this rather complex algorithm is named Tornado, and is used in this project to simulate the aircraft aerodynamics and determine the aerodynamic coefficients of interest.

Since the VLM method don't estimate drag except induced drag, which is directly dependent on lift, we use the wind tunnel data to model the effects of drag and fuselage. Instead when the VLM result is free of dampening from drag, it gives better insight into and understading of the dependencies of the aerodynamic coefficients.

The VLM method is also disregarding the fuselage, which have a great influence on specific parts of the aerodynamic coefficients. Due to the lee of the fuselage. For instance the influence on the vertical stabilizer is damped from changes in angle of attack, because the fuselage force the air to move in a different path around the fuselage.

We have used the VLM simulations in this project to get a more detailed and practical insight into aerodynamics.

5.1.2 Wind Tunnel Testing

The most accurate way to determine the aerodynamics coefficients of an object, is experimental wind tunnel testing.

Since the aerodynamic coefficients are normalised to the wing area S, wing span b and mean chord$c$, it is possible to reuse wind tunnel data from a real size Cessna Skylane 182. The only demands are that the right reference length for the model aircraft are used in (5.1) and (5.2).

Since wind tunnel data includes all aerodynamics effects it is more acurate to use. Thus this project uses the wind tunnel data to fit the coefficients to.

Since there to some coefficients only consists of two samples, is the VLM result used to verify, that it agree with the wind tunnel data[14], and that there is a linear relation in these cases.

5.1.3 Aerodynamic Coefficients

There are six aerodynamic coefficients describing the aerodynamics of an aircraft, all depending on various inputs: relative wind speed V_∞, angle of attack α, side slip angle β, angular rates ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ and control deflection angles: $\boldsymbol{\delta}$. See table 5.1. The coefficients in table 5.1 are found through a combination of the VLM, and wind tunnel data. To reduce the online calculation it is chosen to use simple polynomial fits to represent the coefficients.

It is noted from table 5.1 that the force coeffients are given in the aerodynamics frame $\mathbb{A}$ and

[3] An Induced flow is a flow caused by another flow, just like an induced current from an electromagnetic field.

Coefficient name	Nomenclature
Drag	$C_{\mathbb{A}F_x}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty)$
Side force	$C_{\mathbb{A}F_y}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty)$
Lift	$C_{\mathbb{A}F_z}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty)$
Rolling torque	$C_{\mathbb{B}\tau_x}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty)$
Pitching torque	$C_{\mathbb{B}\tau_y}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty)$
Yawing torque	$C_{\mathbb{B}\tau_z}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty)$

Table 5.1: *The standard aerodynamic coefficients and there dependencies*

the torques are given in the body frame $\mathbb{B}$. This is the standard representation of aerodynamic coefficients since it ease the experiment process, and because the definition of the aerodynamic force tells that they are aligned with the wind direction. The aerodynamic torque is in relation to the body axes of an aircraft.

Before analysing the VLM simulation result and wind tunnel data, we look at the aerodynamic coefficients, and set up a goal for the analysis. We want to find the six coefficients for the Cessna skylan 182 model aircraft and to ease the task on linearizing the model later on we want to find the coefficient on the following form.

$$C_{\mathbb{A}\boldsymbol{F}}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty) = C_{\mathbb{A}\boldsymbol{F}_{zero}}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},V_\infty) + C_{\mathbb{A}\boldsymbol{F}_{input}}(\alpha,\beta,\boldsymbol{\delta})\boldsymbol{\delta} \tag{5.3}$$

$$C_{\mathbb{B}\boldsymbol{\tau}}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},\boldsymbol{\delta},V_\infty) = C_{\mathbb{B}\boldsymbol{\tau}_{zero}}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},V_\infty) + C_{\mathbb{B}\boldsymbol{\tau}_{input}}(\alpha,\beta,\boldsymbol{\delta})\boldsymbol{\delta} \tag{5.4}$$

where $C_{\mathbb{A}\boldsymbol{F}_{zero}}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},V_\infty)$ and $C_{\mathbb{B}\boldsymbol{\tau}_{zero}}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}},V_\infty)$ are the zero dynamic terms with respect to the control inputs, and $C_{\mathbb{A}\boldsymbol{F}_{input}}(\alpha,\beta,\boldsymbol{\delta})$ and $C_{\mathbb{B}\boldsymbol{\tau}_{input}}(\alpha,\beta,\boldsymbol{\delta})$ are control input influence coefficient.

5.1.4 Drag Coefficient

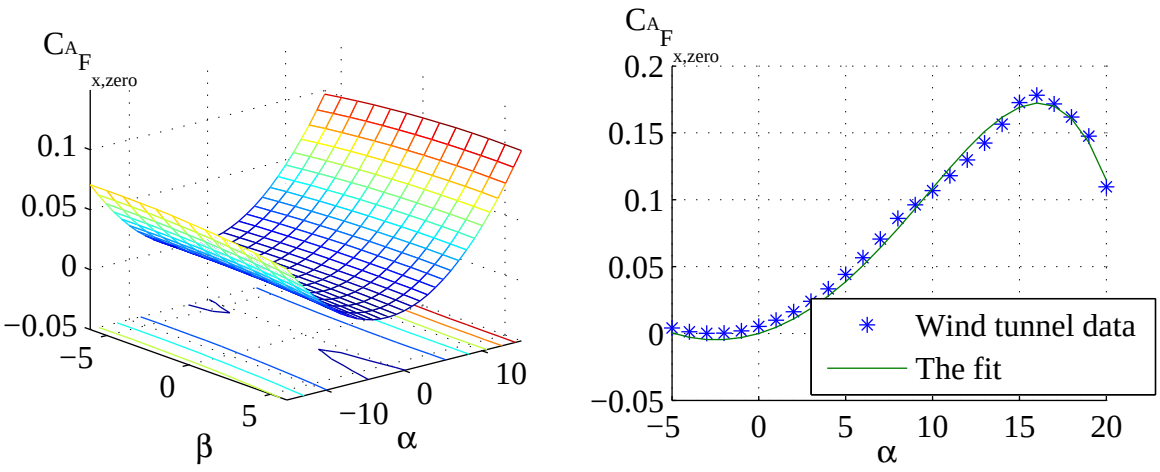
The drag coefficient is a coefficient with a big complexity, due to the various parameters causing drag: skin friction, fuselage, boundary layer effects[4], turbulence and lift.

Since the VLM simulation only takes the lifting influence (induced drag) into account it is a very pour method to represent drag in general. Instead we will fit the drag coefficient to the wind tunnel data, which includes all the effects above.

Zero Dynamics

To find the zero dynamic part of the drag coefficient, we will look at the part of the drag data, which is independent on the control inputs. Thus looking at dependencies regarding to angle of attack and sideslip. The reason for not looking at the angular velocities ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ is that the resulting influence on drag is neglegible. We will use the wind tunnel data to fit the drag coefficient, and use VLM result to show that the valid range of the wind tunnel data is larger than given in the wind tunnel data Figure 5.3 shows both differences and equalities between the wind tunnel data

[4]Boundary layer effects are the aerodynamic effects caused by the difference in air velocity, in different distances from the surface of the aircraft. The air particles just on top of the surface are not moving at all.



(a) VLM simulation of the Drag coefficient.

(b) Wind tunnel data from a real Cessna 182 Skylane [14].

Figure 5.3: *Data used to determine the zero-dynamics Drag coefficient* $C_{A_{F_{x,zero}}}(\alpha, \beta)$.

and the VLM. At high angles we see that figure 5.3(b) includes turbulence or stall effect on the drag coefficient, which isn't included in the VLM result on figure 5.3(a). It is seen that at low α, the induced drag term dominates the drag coefficient, thus the range of the coefficient is extendable in the negative angle of attack direction $\alpha \geq -10°$.

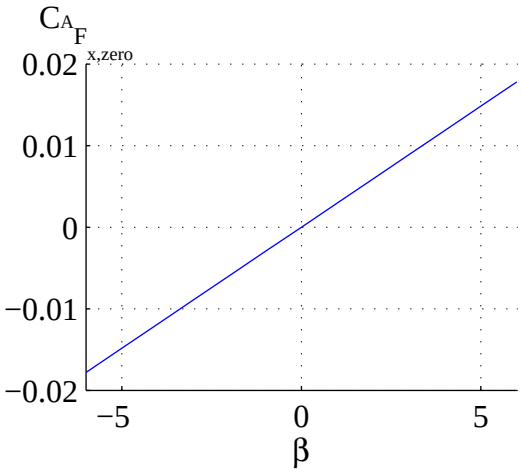


Figure 5.4: *Wind tunnel data the β drag relation.*

The side slip relation on figure 5.4 is also seen on the contur curves of figure 5.3(a), thus this dependecy is taken into account in the aircraft model.

To fit the coefficients we use a least square fit of the wind tunnel data. By using the wind tunnel data instead of the VLM we have achieved to extend the detail of the model by including stall at high angle of attacks $\alpha \geq 15°$.

The least square fit for the zero dynamics is found using the following polynomial candidate (5.5).

$$C_{\mathbb{A}F_{x,zero}}(\alpha,\beta) = C_{\mathbb{A}F_{x,0}} + \boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{x,zero}}(\alpha)\begin{bmatrix}\alpha & \beta\end{bmatrix}^{\mathrm{T}} \tag{5.5}$$

The candidate function is found through an iteration of different candidates for the VLM data, until the best candidate, resulting in the lowest fit error variance, where found see (5.6). The reason for not having the α^3 term is that $C_{\mathbb{A}F_{x,zero}}$ wound have a to flat slope.

The zero dynamics parameter vector are fitted to (5.6), and the constant $C_{\mathbb{A}F_{x,0}} = 0.027$ represents the skin friction.

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{x,zero}}(\alpha) = \begin{bmatrix}0.22211 + 2.5785\alpha - 53.4504\alpha^4 & 0.17\end{bmatrix} \tag{5.6}$$

Control Input Dynamics

We look at the control inputs influence on the drag coefficient, in the same way as for the zero dynamics. One issue here is to identify which control surfaces influences the drag coefficient. It turns out to be the flaps of the aircraft since it is the only one with a non-negligible effect. Even though this is out of the scope of this project we include this into the model to ease future projects.

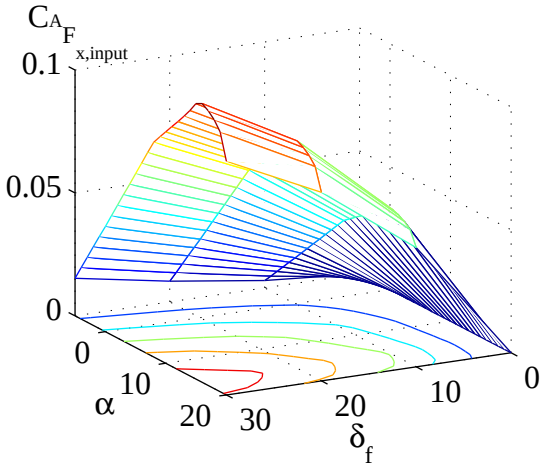


Figure 5.5: *Wind tunnel data of flap influence on drag from a real Cessna 182 Skylane [14].*

The retalion between flap control input δ_f, and α to the drag coefficient is shown on figure 5.5. We don't use the VLM data at this point since it don't handle drag simulation as turbulence which is the main cause for the drag change on the aircraft due to flap deflection.

The least square candidate used to fit the control influence is (5.7).

$$C_{\mathbb{A}F_{x,input}}(\alpha,\boldsymbol{\delta}) = \boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{x,input}}(\alpha)\delta_f \tag{5.7}$$

The control input influence coefficient is then fitted to (5.7), where δ_f is the flap deflection angle and the result of the least square fit of the input coefficient is (5.8).

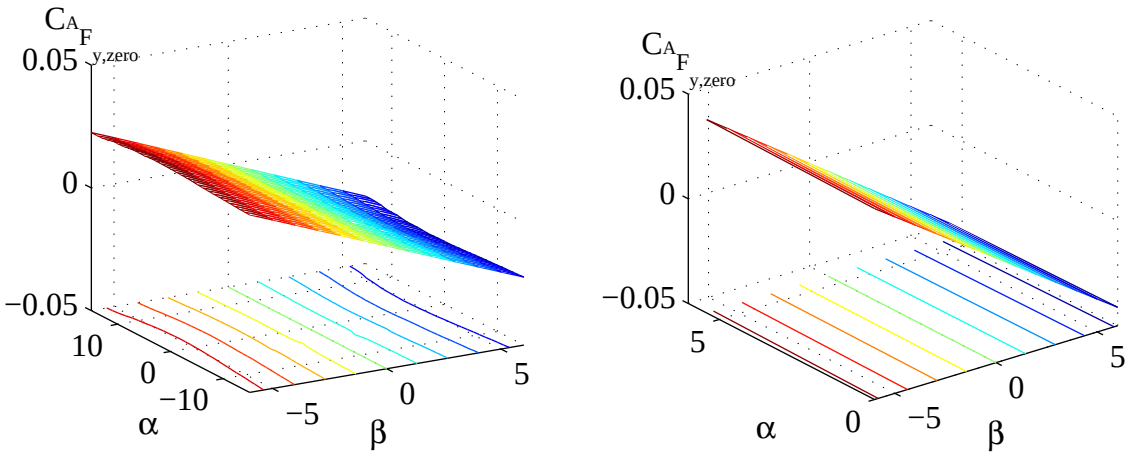
$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{x,input}}(\alpha) = \begin{bmatrix}0.085124 + 0.6536\alpha - 0.08006\alpha^2 - 20.2106\alpha^5\end{bmatrix} \tag{5.8}$$

5.1.5 Side force Coefficient

The side force coefficient $C_{\mathbb{A}F_y}$ is found in a similar way as the drag coefficient. To find the important dependencies of the side force coefficient, we look at how side force is generated. Basicly side force is caused by sideslip β and yawing motion ${}^{\mathbb{B}}\omega_{\mathbb{B},z}$, thus the rudder control surface causes side force. Another important parameter is the roll rate ${}^{\mathbb{A}}\omega_{\mathbb{B},x}$, because when the aircraft rotates the air flow direction change, this could cause the air to flow not over the wing, but along the span of the wing. Thus creating a side force.

Zero Dynamics

In this section we will fit the wind tunnel data to an expression for the zero dynamics side force coefficient $C_{\mathbb{A}\boldsymbol{F}_{y,zero}}(\beta, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}, V_\infty)$.



(a) VLM simulation of the side force coefficient.

(b) Wind tunnel data for the side force coefficient with respect to sideslip and angle of attack.

Figure 5.6: *Data used to fit the zero dynamics coefficient with respect to sideslip.*

The wind tunnel data on figure 5.6(b) shows that there is no dependency with regard to the angle of attack.

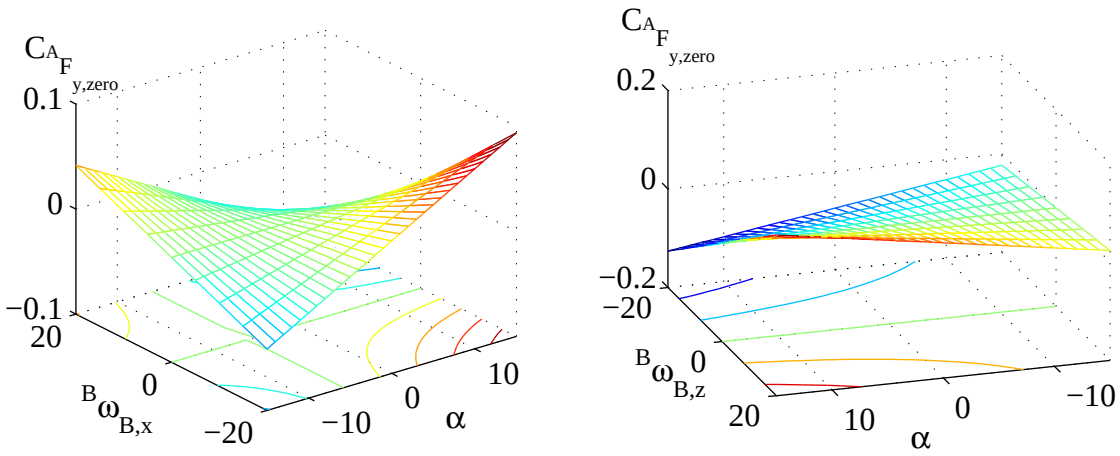
The dependecies with regard to ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ is taken from figure 5.7(a) and 5.7(b).

The resulting zero dynamic coefficient polynomial (5.9) is found as,

$$C_{\mathbb{A}F_{y,zero}}(\alpha, \beta, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}, V_\infty) = \boldsymbol{P}^{\mathrm{T}}_{{}^{\mathbb{A}}F_{y,zero}}(\alpha)\begin{bmatrix}\beta & \frac{b}{2V_\infty}{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}\end{bmatrix}^{\mathrm{T}} \tag{5.9}$$

where $\frac{b}{2V_\infty}$ is a unit normalization factor consisting of: $b = 2.405$m the reference wing span, and V_∞ the relative wind speed. This is important because then the coefficient is still dimension less[5].

[5]The dimension less property of the coefficients, is used such that the coefficients don't depends on the size



(a) Wind tunnel roll rate relation to the side-force.

(b) Wind tunnel yaw rate relation to the side-force.

Figure 5.7: *The angular rate relation to the aerodynamic side-force coefficient.*

The least square solution vector is (5.10).

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{y,zero}}(\alpha) = \begin{bmatrix} -0.393 - 0.11665\alpha & -0.075 - 0.74231\alpha & 0 & 0.214 + 0.56204\alpha \end{bmatrix} \tag{5.10}$$

Control Input Dynamics

The control input direct influence on side force is restricted to the rudder.

The wind tunnel data, together with the VLM result shows the linear relation between control surface deflection,δ_r and side force coefficient shown on figure 5.8. As shown on figure 5.8 there is a linear relation between the coefficient and the control input. From the VLM data we see by looking at the curvatures spreading see figure 5.8(a), that there is a cross relation between α and δ_r, but in the wind tunnel date this effect is cancelled out, since the fuselage have an effect on the airflow around the vertical stabilizer at different angle's of attack.

We fit the coefficient on the following form (5.11) using the wind tunnel data since it properly is more accurate than the VLM.

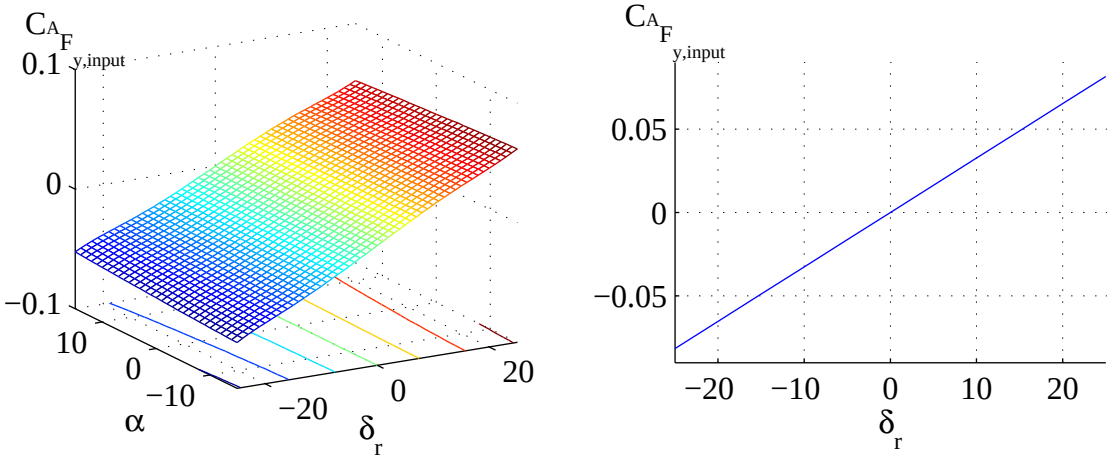
$$C_{\mathbb{A}F_{y,input}}(\boldsymbol{\delta}) = \boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{y,input}}\boldsymbol{\delta} \tag{5.11}$$

the resulting least square solution is (5.12),

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{y,input}} = \begin{bmatrix} 0 & 0 & 0 & 0.187 \end{bmatrix} \tag{5.12}$$

where the control input vector is defined as $\boldsymbol{\delta} = \begin{bmatrix} \delta_f & \delta_a & \delta_e & \delta_r \end{bmatrix}^{\mathrm{T}}$.

and length of the aircraft only the shape. Instead special reference lengths b and c are used to scale the size of the aircraft.



(a) VLM simulation of the side force coefficient, with respect to the control surfacedeflection angle δ_r.

(b) Wind tunnel data for the side force coefficient with respect to control surface deflection δ_r.

Figure 5.8: *Data used to fit the control input coefficient with respect to rudder deflection.*

5.1.6 Lift Coefficient

The Lift coefficient like the two previous is found through least square approximation. The Lift coefficient differs from the two other forces because it has a dependency toward a new parameter, namely $\dot{\alpha}$ the time derivative of the angle of attack, as well as the pitching rate.

First we will find the zero dynamics followed by the control influence from the two input flaps and elevator.

Zero Dynamics

As for the drag coefficient the lift coefficients range is extended, since the VLM simulation have the same tendencies as the wind tunnel data. So by combining the two results on figure 5.9, we extend the range of α such that $-10° \leq \alpha \leq 22°$. And once again we get stall effects into our model by using this approach in stead of only relying on VLM simulations.

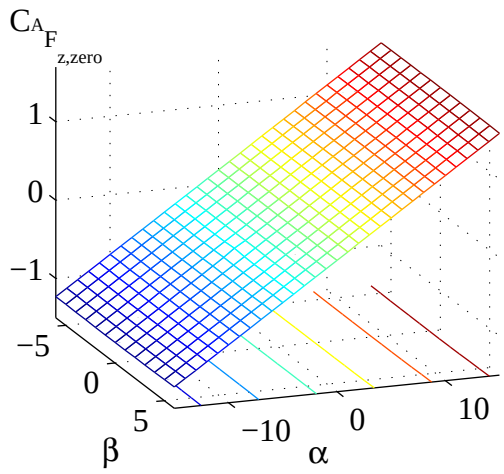
The data on figure 5.9, represents the part of the zero dynamics lift coefficient, which needs to be fitted to a polynomial.

The remaining part is shown on figure 5.10, where the pitch angular rate ${}^{\mathbb{B}}\omega_{\mathbb{B},y}$ relation is shown, together with the relative wind speed direction change $\dot{\alpha}$.

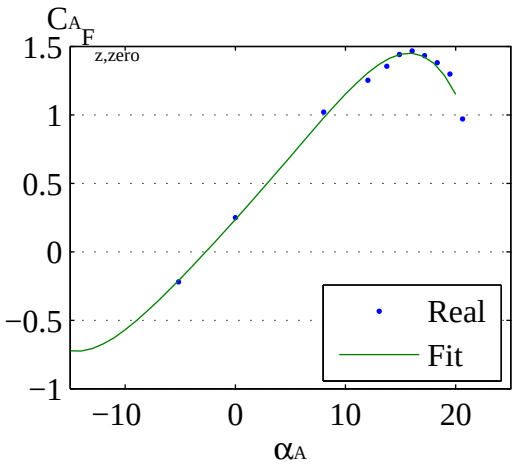
To fit the part of the zero dynamics represented on figure 5.9(b) and 5.8(a), we use the form of (5.13) in the least square fit, and we represent the angles in radians.

$$C_{\mathbb{A}F_{z,zero}}(\alpha, \dot{\alpha}, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}, V_\infty) = C_{\mathbb{A}F_{z,0}} + \boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{z,zero}}(\alpha)\left[\alpha \quad \frac{c}{2V_\infty}\dot{\alpha} \quad \frac{c}{2V_\infty}{}^{\mathbb{B}}\boldsymbol{\omega}^{\mathrm{T}}_{\mathbb{B}}\right]^{\mathrm{T}} \tag{5.13}$$

In (5.13) the values $c = 0.336$m is the reference chord, and $\frac{c}{2V_\infty}$ is a unit normalization factor



(a) VLM simulation of the lift coefficient, with respect to the angle of attack.



(b) Wind tunnel data for the lift coefficient, with respect to the angle of attack. And the least square fit of the data points.

Figure 5.9: *Data used to fit the zero dynamics coefficient with respect to angle of attack.*

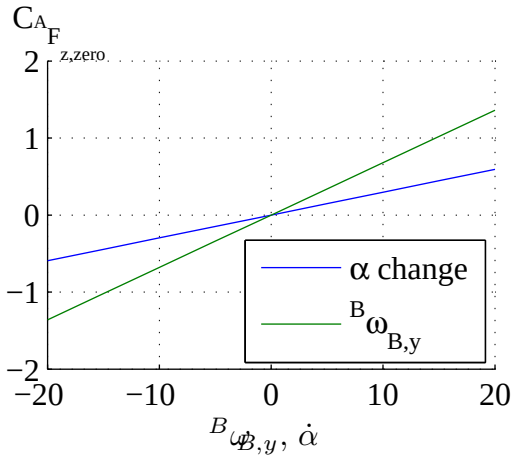


Figure 5.10: *The angular rate relation to the lift coefficient.*

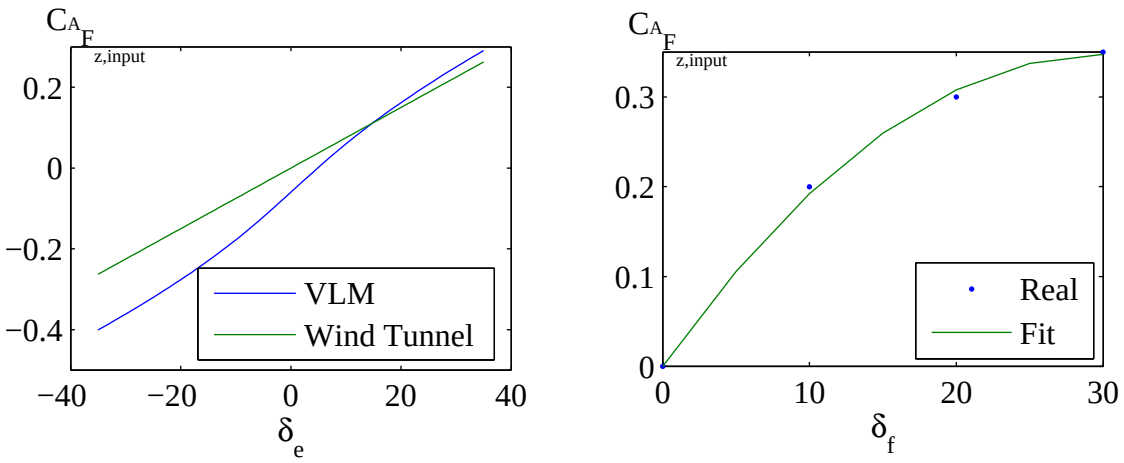
for the velocity parts of the coefficient, as in the side force coefficient see section 5.1.5. $C_{\mathbb{A}F_{z,0}} = 0.307$ is the lift coefficient at $\alpha = 0$.

The least square solution for (5.13) is found as (5.14).

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{z,zero}}(\alpha) = \begin{bmatrix} 5.3446 - 0.78182\alpha - 177.7791\alpha^4 & 1.7 & 0 & 3.9 & 0 \end{bmatrix} \tag{5.14}$$

Control Input Dynamics

The control surfaces influencing on lift is the flaps and elevator. To find the control influence coefficient we look at these input effects.



(a) The control influence from the Elevator, both data sets from the VLM simulation and the Wind tunnel data.

(b) The control influence from the flap, based on the wind tunnel data.

Figure 5.11: *Data used to fit the control input influence coefficient with respect to the control surface deflection angles for flaps δ_f and elevator δ_e.*

Figure 5.11 shows the dependencies of the lift coefficient with regard to the control input δ_f, and δ_e.

From figure 5.11(a) we see that the dependency to δ_e is linear, when looking at the wind tunnel data. And from figure 5.11(b) we see that there is a higher order dependency with regard to δ_f. Trying different polynomial candidates found that (5.15) is suitable for fitting a polynomial to the lift control input coefficient.

$$C_{\mathbb{A}F_{z,input}}(\boldsymbol{\delta}) = \boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{z,input}}(\delta_f)\boldsymbol{\delta} \tag{5.15}$$

The parameter vector is determind to (5.16),

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{z,input}}(\delta_f) = \begin{bmatrix} 1.4591 - 1.8055\delta_f + 2.0582\delta_f^4 & 0 & 0.43 & 0 \end{bmatrix} \tag{5.16}$$

when the control input vector is defined as $\boldsymbol{\delta} = \begin{bmatrix} \delta_f & \delta_a & \delta_e & \delta_r \end{bmatrix}^{\mathrm{T}}$.

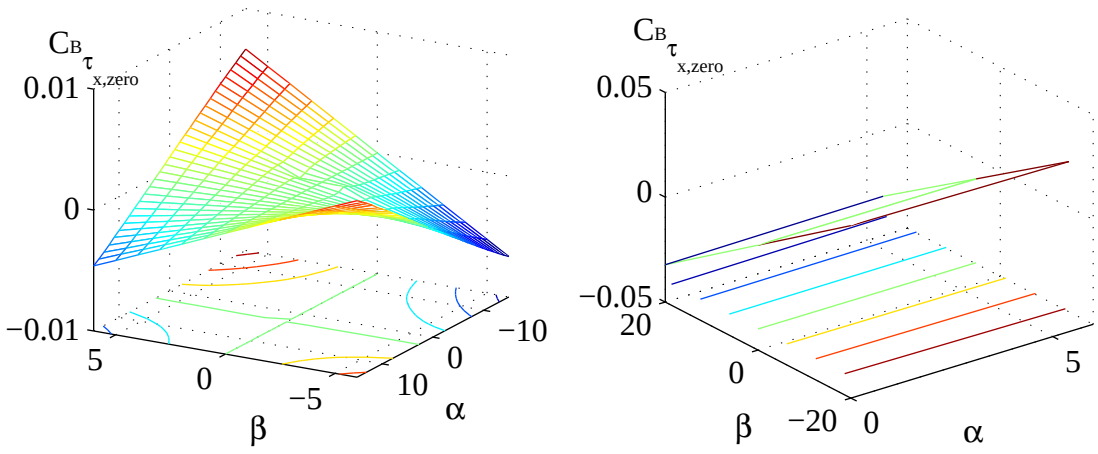
5.1.7 Roll Torque Coefficient

As with the force coefficient there exist torque coefficient, these are found using the same method. We look at the zero dynamics part followed by the control influence part.

The roll torque is caused by a asymmetric pressure distribution on the aircraft around the $x_{\mathbb{B}}$ axis. Thus the general influences is caused by: Sideslip β, Aileron deflection δ_a, and rudder deflection δ_r. Another very important point to notice is that angular velocities also have a great effect on the roll coefficient. Actually this is a natural damping of the aircraft, since the pressure change caused by the angular velocity of the aircraft work opposite the torque generated by the deflection of control surfaces. Thus the two angular velocities: ${}^{\mathbb{A}}\omega_{\mathbb{B},x}$, and ${}^{\mathbb{A}}\omega_{\mathbb{B},z}$ are influencing the roll coefficient.

Zero Dynamics

The zero dynamics is found through the VLM result combined with the wind tunnel data. We will fit the polynomial to the wind tunnel data, and use the VLM simulation to extend the working range of the coefficient to $-10° \leq \alpha \leq 22°$. The two figures 5.12, shows the zero



(a) The VLM result showing the relation with respect to: β, and α.

(b) The wind tunnel data showing the relation with respect to: β, and α.

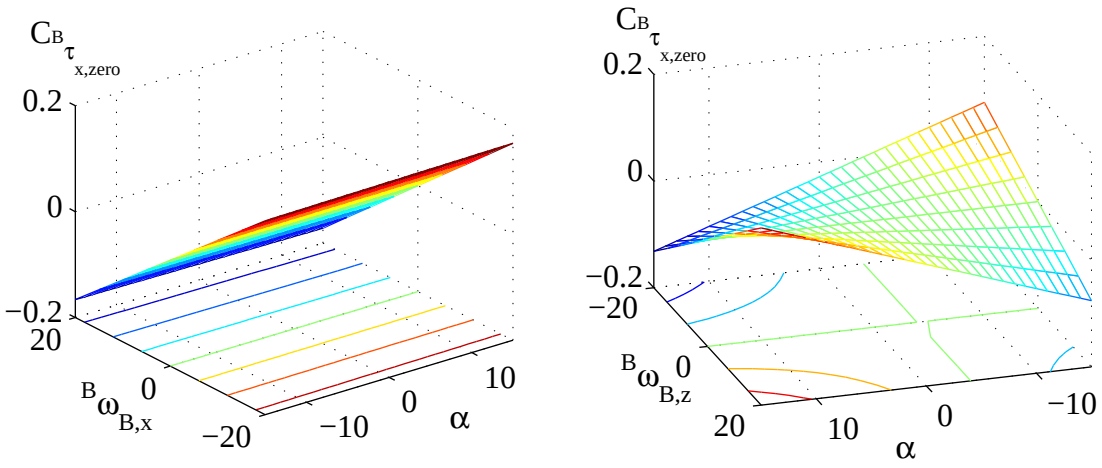
Figure 5.12: *Wind tunnel data and VLM simulation result used for determination of the Roll zero dynamics coefficient.*

dynamic part of the Roll coefficient. The first look at the graphs tells us that they are very different, but actually they have equal form, but they show exactly, what the missing details in VLM simulations have. If we look carefully at the curves on the β and α plane on figure 5.12(b), we notice that they have a tendency to move away from $\beta = 0$, when α increases, thus the equation representing the coefficient have the following dependencies: β, and $\beta\alpha$. When looking at figure 5.12(a) we see that it have the same form especially the $\beta\alpha$ part is shown. This tells

us that the part, where α have influence, is damped in the wind tunnel data. This is because the wind tunnel data includes the fuselage.

The reason for the zero dynamics in the roll coefficient is the vertical stabilizer. The side force of the vertical stabilizer airfoil perpendicular to $x_{\mathbb{B}}$ changes with the sideslip angle β. The roll moment is caused since there only is a vertical stabilizer on upper side of the fuselage. The damping from the fuselage is because the vertical stabilizer is in the lee of the fuselage.

One stabilising factor of an aircraft is the natural damping from the angular rates influencing the airflow, as mentioned in section 5.1.5. From the wind tunnel data we get the rolling and yawing influence data shown on figure 5.12.



(a) The wind tunnel data showing angular roll rate ${}^{\mathbb{A}}\omega_{\mathbb{B},x}$ relation with respect to the cross relation of α.

(b) The wind tunnel data showing angular yaw rate ${}^{\mathbb{A}}\omega_{\mathbb{B},z}$ relation with respect to the cross relation of α.

Figure 5.13: *Wind tunnel data and VLM simulation result used for determination of the damping part of the Roll zero dynamics coefficient.*

From figure: 5.13(a), and 5.13(b) we get the damping relation to the roll coefficient. Together with the α and β relation from figure 5.12 we get following equation for the roll torque coefficient.

$$C_{\mathbb{B}\tau_{x,zero}} = \boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{x,zero}}(\alpha)\begin{bmatrix}\beta & \frac{b}{2V_\infty}{}^{\mathbb{B}}\omega_{\mathbb{B},x} & \frac{b}{2V_\infty}{}^{\mathbb{B}}\omega_{\mathbb{B},z}\end{bmatrix}^{\mathrm{T}} \tag{5.17}$$

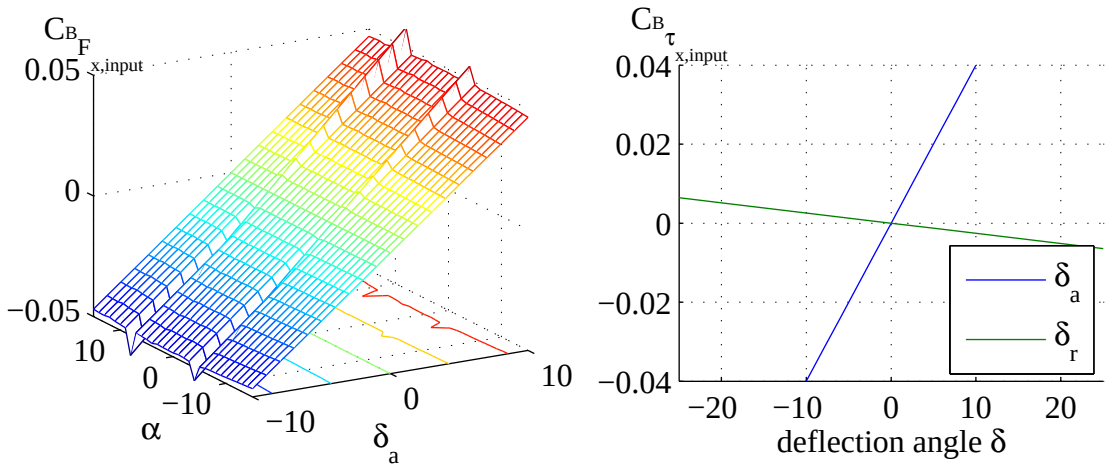
The least square fitted parameter vector in (5.17) is found to (5.18)

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{x,zero}}(\alpha) = \begin{bmatrix}-0.092264 + 0.030385\alpha & -0.484 - 0.031813\alpha & 0.0798 + 1.1357\alpha\end{bmatrix} \tag{5.18}$$

Control Input Dynamics

The control surfaces influencing the roll coefficient is both the Aileron and the rudder. We use the wind tunnel data to find a suitable expression for the coefficient and look at the VLM

results tendencies to verify that the working area of the coefficients are expandable, since the data shown on figure 5.14(b) only consists of 4 samples in all.



(a) The VLM result showing the relation with respect to: δ_a.

(b) The wind tunnel data showing the relation with respect to: δ_a.

Figure 5.14: *Wind tunnel data and VLM simulation result used for determination of the Roll zero dynamics coefficient.*

The control input influence from aileron is shown on the figure 5.14(a). Which shows an almost linear relation between δ_a and the coefficient, which confirms the wind tunnel data. The anomalies on figure 5.14(a) are due to numerical differences. As shown on figure 5.15 there is a linear relation between the rudder and the roll coefficient as for the aileron. Thus the polynomial candiate for the least square fit is:

$$C_{\mathbb{B}_{\tau_x,input}}(\boldsymbol{\delta}) = \boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}_{\tau_{x,input}}} \begin{bmatrix} \delta_a & \delta_r \end{bmatrix}^{\mathrm{T}} \tag{5.19}$$

The least square solution for the parameter vector in (5.19) is (5.20).

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}_{\tau_{x,input}}} = \begin{bmatrix} 0.229 & -0.0147 \end{bmatrix} \tag{5.20}$$

It is noted that the small dependency with respect to α on figure 5.14(a) and 5.15 is negligible.

5.1.8 Pitch Torque Coefficient

The Pitch torque is caused because of the pressure distribution chord-wise[6] an aerofoil. Since the elevator deflection δ_e changes the pressure around the aircraft tail, this will cause an pitching torque, similarly is the flaps δ_f and aileron δ_a having an influence, but since the aileron surface area chord-wise is very small it do not make a noticeable pressure change, thus we disregard the aileron deflection. The air velocity changes along the chordline of an airfoil influences the pressure distribution, thus α, $\dot{\alpha}$, and the pitching angular rate ${}^{\mathbb{B}}\omega_{\mathbb{B}}$ are important dependencies.

[6] chord-wise is the path over the airfoils along the $\boldsymbol{x}_{\mathbb{B}}$ axis

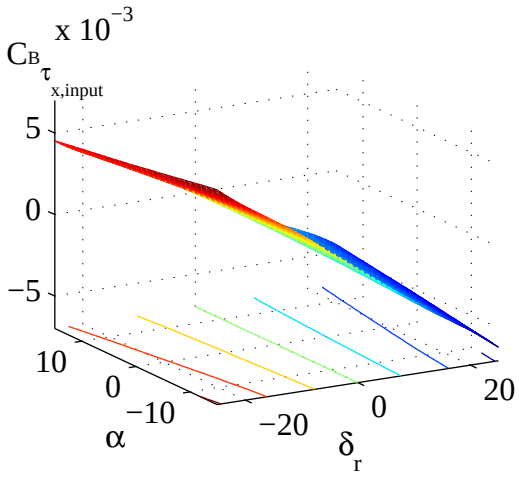
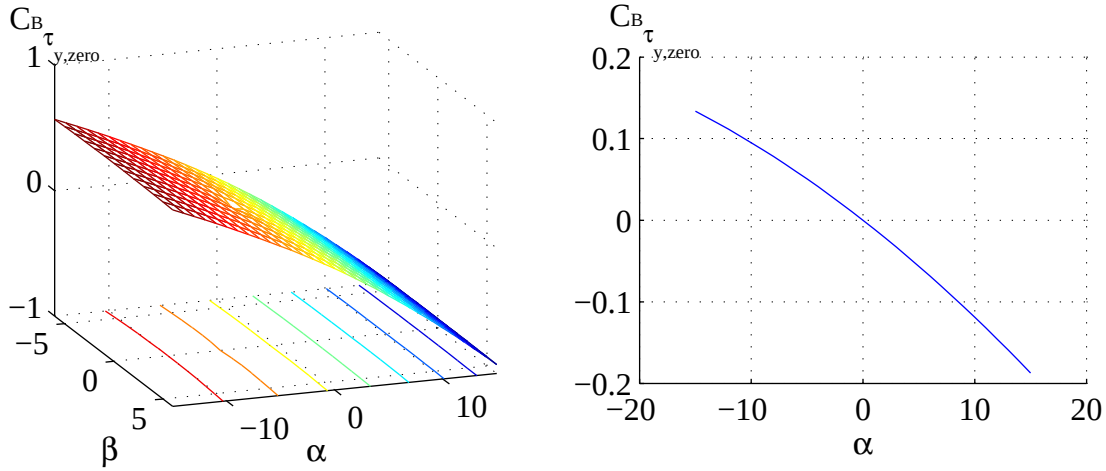


Figure 5.15: *The VLM result showing the relation with respect to: δ_r.*

Zero Dynamics

First, we look at the dynamics not influenced by the control inputs, namely the α, $\dot{\alpha}$ and ${}^{\mathbb{B}}\omega_{\mathbb{B}}$. On figure 5.16 we see that there is a nonlinear dependency with regard to α, And it is noticed



(a) The VLM result showing the relation with respect to: β, and α.

(b) The wind tunnel data showing the relation with respect to α.

Figure 5.16: *The Pitch-Torque coefficient dependencies.*

that a positive angle of attack results in a negative torque. Thus the aircraft is stabilised through the pitch torque.

Another influence on the pitch torque is the rate of which the wind direction change. This part isn't included in the VLM simulations because it actually is a drag effect.

When looking at figure 5.17 we see that if the aircraft change orientation very fast, it changes

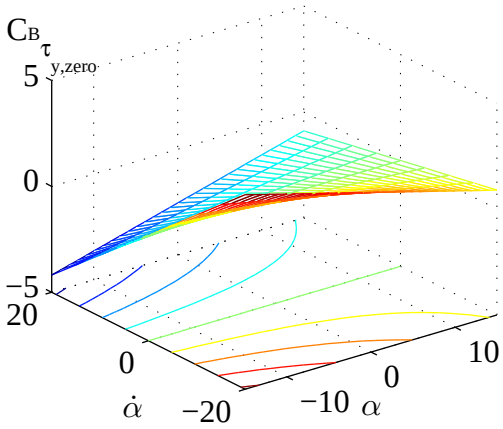


Figure 5.17: *Pitch Torque coefficient dependency, with regard to the change in wind direction $\dot{\alpha}$ and its cross relation to α.*

its pitching torque. When the aircraft moves upward, air is forced to move around the aircraft downward causing a higher pressure on top of the aircrafts nose[7]. In fact this also helps reducing the pitch torque of the aircraft, since it works against the applied torque.

A similar velocity acting like $\dot{\alpha}$, is the pitch rate ${}^{\mathbb{B}}\omega_{\mathbb{B},y}$. The reason for dealing with both of them individually even though they seems equal is that, the aircraft could fall down while pitching upward, thus α and ${}^{\mathbb{B}}\omega_{\mathbb{B},y}$ cancels each other out.

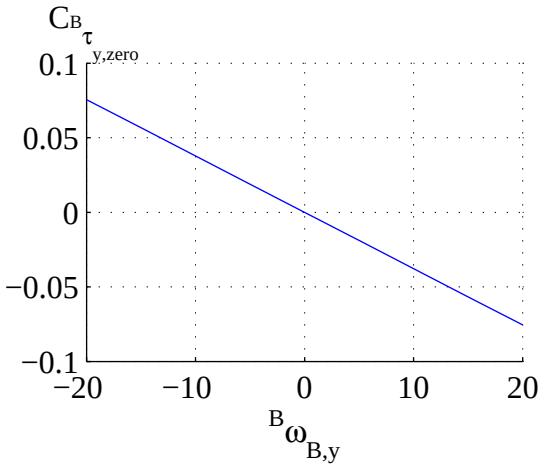


Figure 5.18: *The Pitch-Torque coefficient dependency with regard to the pitch rate ${}^{\mathbb{B}}\omega_{\mathbb{B},y}$, from the wind tunnel dataset.*

Looking at figure 5.18, we see that just like the two terms above this influence is damping the pitch torque.

[7]The same effect happens when a piece of paper is move fast upward through the air, it bends downward, since the air is moving downwards.

The least square candidate function we used to fit this part of the yaw torque coefficient with is (5.21),

$$C_{\mathbb{B}_{\tau_{y,zero}}}(\alpha, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}) = C_{\mathbb{B}_{\tau_{y,0}}} + \boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}_{\tau_{y,zero}}}(\alpha)\begin{bmatrix}\alpha & \dot{\alpha} & \frac{b}{2V_\infty}{}^{\mathbb{B}}\omega_{\mathbb{B},y}\end{bmatrix}^{\mathrm{T}} \tag{5.21}$$

where the fitted polynomial parameters are (5.22), $C_{\mathbb{B}_{\tau_{y,0}}} = 0.04$.

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}_{\tau_{y,zero}}}(\alpha) = \begin{bmatrix}-0.613 - 0.39236\alpha & -7.27 + 18.0276\alpha & -12.4\end{bmatrix} \tag{5.22}$$

Control Input Dynamics

The first control input influence on the pitch torque coefficient, is the flaps. Looking at the flap influence on figure 5.19, we see one of the purposes of flaps; damping the constant pitch torque $C_{\mathbb{B}_{\tau_{y,0}}}$. But unfortunately it is also seen that it is possible to give the aircraft to much flap, such that it stalls. Another control input is the elevator δ_e, which also is the more direct pitch control

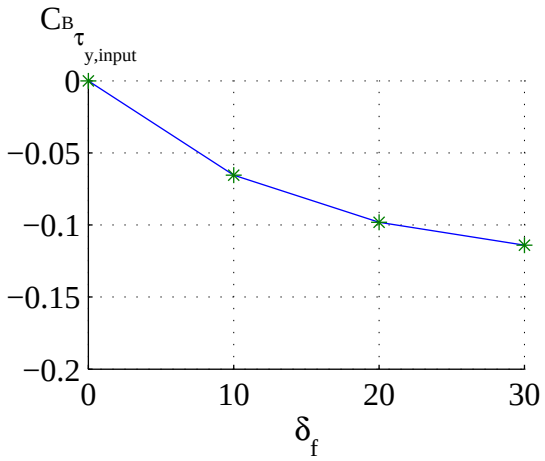


Figure 5.19: *The flap relation to the pitching torque.*

input. This is seen because it is the only control surface, see figure 5.20, which is able to force both negative and positive pitch torque. It is also noticed that on figure 5.20 there is a cross relation α and δ_e, which is typical for this wing because the elevator generates pitch torque due to the lift of the vertical stabiliser. This is also the reason why the aircraft is constructed, such that the vertical stabiliser is a symmetric airfoil, since it only generates lift at $\alpha \neq 0$ or if it is geometrically deformed, such that they isn't symmetric, $\delta_e \neq 0$.

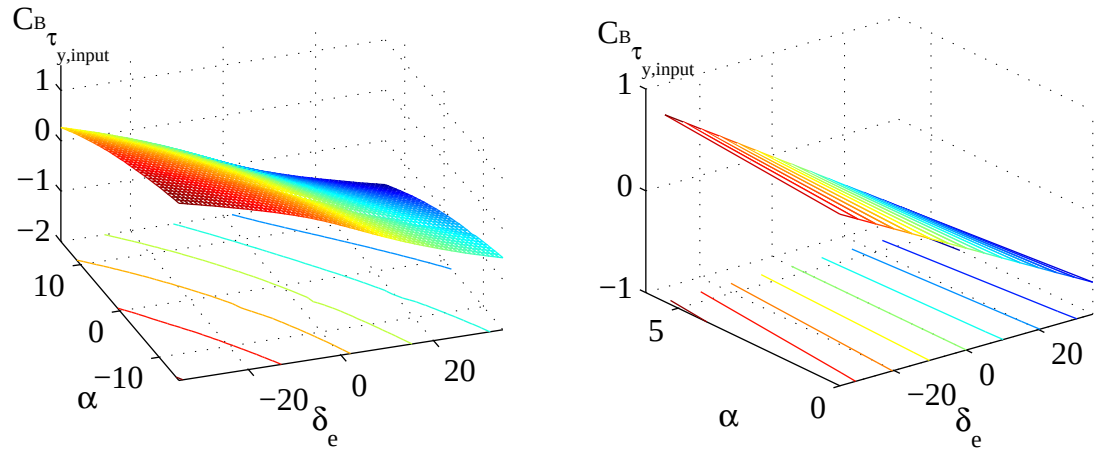
The elevator influence is damped by the fuselage, which blocks the airflow to some degree, this is noted on difference between the curvatures in the α,β plane on both figure 5.20.

The least square candidate function (5.23), is used to fit the input influence term of the pitch torque coefficient.

$$C_{\mathbb{B}_{\tau_{y,input}}}(\alpha, \boldsymbol{\delta}) = \boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}_{\tau_{y,input}}}(\alpha, \boldsymbol{\delta})\boldsymbol{\delta} \tag{5.23}$$

The fitted polynomial parameters are (5.24).

$$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}_{\tau_{y,input}}}(\alpha, \boldsymbol{\delta}) = \begin{bmatrix}-0.4769 + 0.58894\delta_f - 0.65451\delta_f^4 & 0 & -1.122 - 2.6193\alpha & 0\end{bmatrix} \tag{5.24}$$



(a) The VLM result showing the relation with respect to: δ_e, and α.

(b) The wind tunnel data for the elevator deflection angle δ_e.

Figure 5.20: *The elevator influence on the pitching torque.*

5.1.9 Yaw Torque Coefficient

The yawing part of the Torque coefficients are primarily caused by the vertical stabilizer. Since this airfoil only resides on the topside of the fuselage, the arodynamic forces on this airfoil are unevenly distributed around the ${}^{\mathbb{B}}\boldsymbol{x}$ axis, and located at the tail of the aircraft. Thus contributing to the Roll and Yaw torque of the aircraft. Similarly an asymmetric drag distribution over the main wing causes a yaw torque.

Thus the yaw torque coefficient is $C_{{}^{\mathbb{B}}\tau_z}(\alpha, \beta, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}, \boldsymbol{\delta})$, where the dependencies: flap deflection δ_f, elevator deflection δ_e, and pitch rate ${}^{\mathbb{A}}\omega_{\mathbb{B},y}$ has no influence on the coefficient.

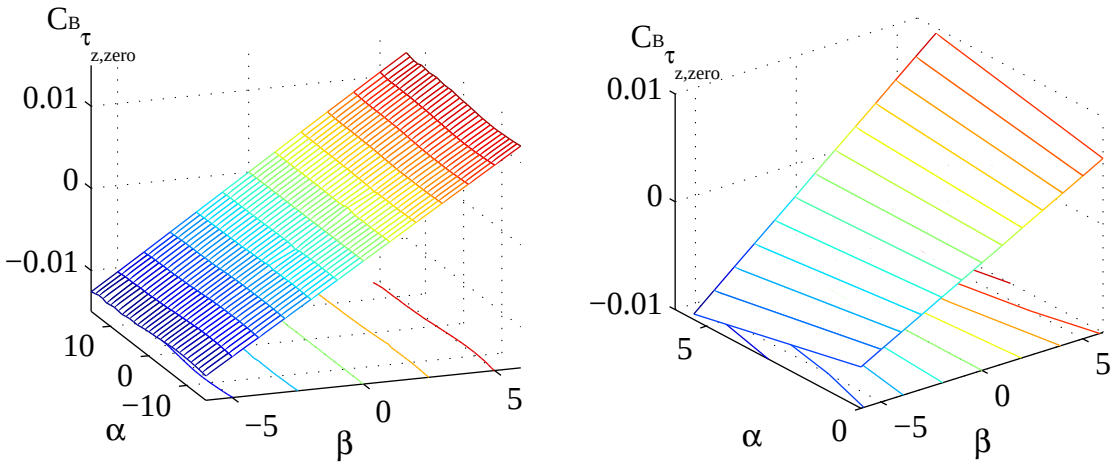
Zero Dynamics

Since the airfoil shape of the vertical stabilizer is symmetric, is the pressure evenly distributed on both sides of the airfoil, thus there are no yaw torque when $\beta = 0$. This also counts for the main wing influence, since the pressure is evenly distributed over the wing, when $\beta = 0$.

As seen on figure 5.21 the yaw-torque coefficient is dependent on β. It should also be noted that it has a positive influence on stability. The β influence is dampening, because a positive sideslip angle results in a positive torque, thus smaller sideslip.

Another dependency is the angular roll rate ${}^{\mathbb{B}}\omega_{\mathbb{B},x}$, which is shown on figure 5.22.

When looking at figure 5.22, we observe that a positive roll rate have a dampening effect on the yaw torque. This is in fact a good stability discovery, because as seen in section 5.1.7 the roll rate is generate because of side-slip, thus the aircraft is constructed in a aerodynamic stable way.



(a) The VLM result showing the relation with respect to: β, and α.

(b) The wind tunnel data showing the relation with respect to: β, and α.

Figure 5.21: *The Yaw-Torque coefficient dependency with regard to α and β,from the wind tunnel dataset.*

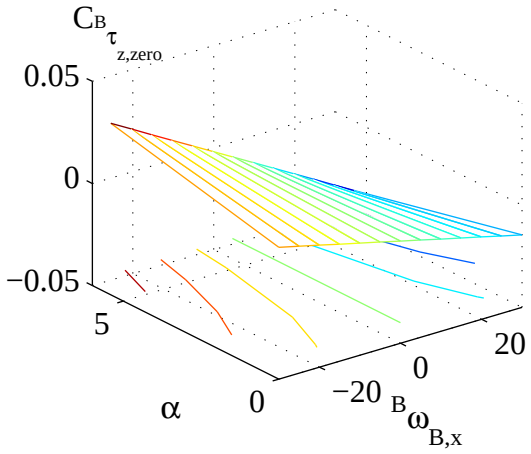


Figure 5.22: *The Yaw-Torque coefficient dependency with regard to ${}^{\mathbb{B}}\omega_{\mathbb{B},x}$, from the wind tunnel dataset.*

Also, when looking at the angular yaw rate ${}^{\mathbb{B}}\omega_{\mathbb{B},z}$ dependency on figure 5.23, we notice the same

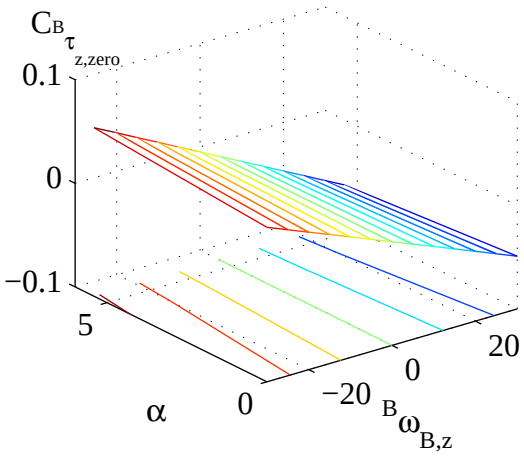


Figure 5.23: *The Yaw-Torque coefficient dependency with regard to ${}^{\mathbb{B}}\omega_{\mathbb{B},z}$, from the wind tunnel dataset.*

dampening effect as for the roll rate.

To model the zero dynamic part of the yaw torque we use a least square fit of the wind tunnel data. The least square candidate function we use to fit this part of the yaw torque coefficient with is (5.25),

$$C_{{}^{\mathbb{B}}\tau_{z,zero}}(\alpha,\beta,{}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}) = \boldsymbol{P}^{\mathrm{T}}_{{}^{\mathbb{B}}\tau_{z,zero}}(\alpha)\begin{bmatrix}\beta & \frac{b}{2V_\infty}{}^{\mathbb{B}}\omega_{\mathbb{B},x} & \frac{b}{2V_\infty}{}^{\mathbb{B}}\omega_{\mathbb{B},z}\end{bmatrix}^{\mathrm{T}} \tag{5.25}$$

where the fitted polynomial parameters are (5.26).

$$\boldsymbol{P}^{\mathrm{T}}_{{}^{\mathbb{B}}\tau_{z,zero}}(\alpha) = \begin{bmatrix}0.0587 + 0.032\alpha & -0.0278 - 0.0371\alpha & -0.0937 - 0.0262\alpha\end{bmatrix} \tag{5.26}$$

Control Input Dynamics

The yaw torque of the aircraft is like all the other aerodynamic states, influenced by the control input especially the aileron deflection δ_a and rudder deflection δ_r, which is covered in this section.

Looking at figure 5.24, we see that there is a nonlinear relation between the α, δ_a, and the yaw torque coefficient. The VLM method isn't used in this case, because the yaw torque generated by the aileron, is caused due to changes in drag over the main wing. Similar to the δ_a influence on figure 5.24, there is a nonlinear relation between: α, δ_a, and the yaw torque on figure 5.25.

The least square candidate function (5.27), is used to fit the input influence term of the yaw torque coefficient.

$$C_{{}^{\mathbb{B}}\tau_{z,input}}(\alpha,\boldsymbol{\delta}) = \boldsymbol{P}^{\mathrm{T}}_{{}^{\mathbb{B}}\tau_{z,input}}(\alpha)\boldsymbol{\delta} \tag{5.27}$$

The fitted polynomial parameters are (5.28).

$$\boldsymbol{P}^{\mathrm{T}}_{{}^{\mathbb{B}}\tau_{z,input}}(\alpha) = \begin{bmatrix}0 & -0.0216 - 0.0288\alpha & 0 & -0.0645 - 0.016\alpha\end{bmatrix} \tag{5.28}$$

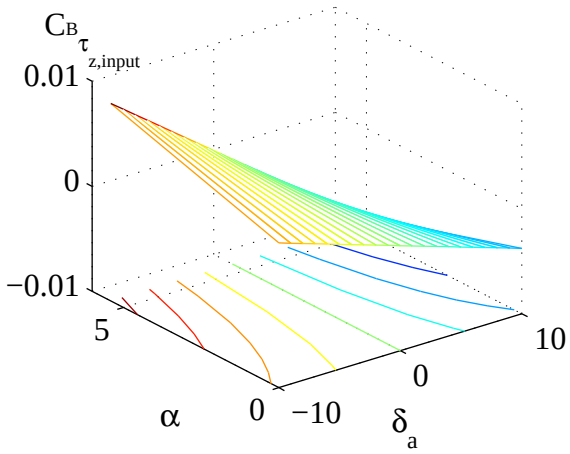
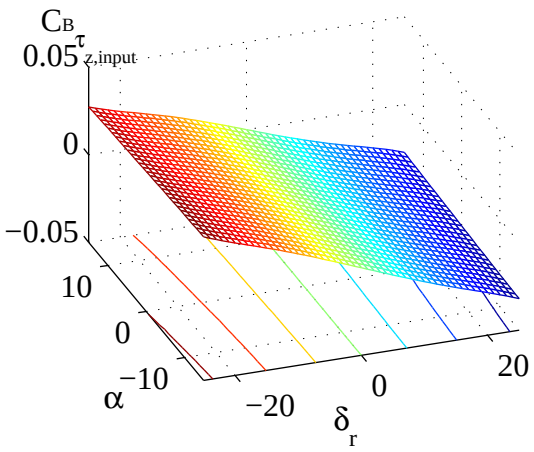
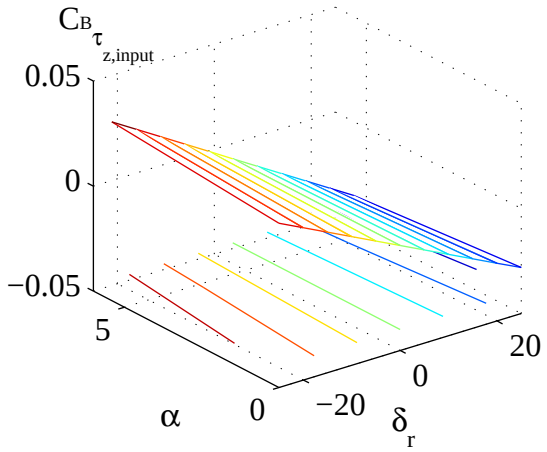


Figure 5.24: *The wind tunnel data showing the relation with respect to:* δ_a.



(a) The VLM result showing the relation with respect to: δ_r.



(b) The VLM result showing the relation with respect to: δ_r.

Figure 5.25: *The Yaw-Torque coefficient dependency with regard to* δ_r*, from the wind tunnel dataset.*

5.2 Propeller

This section contains a model of the propeller. This model is derived using an implementation of the Glauret's Blade Element Theory (BET) to determine the thrust ${}^{\mathbb{B}}\boldsymbol{F}_P$ and torque ${}^{\mathbb{B}}\boldsymbol{\tau}_P$ at a given V_∞ and angular velocity of the propeller $\boldsymbol{\omega}_P$. It is chosen to use JavaProp [5], which is a ready made implementation of BET.

5.2.1 Blade Element Theory

When the propeller is rotating it accelerate the air. Following Newtons third law this produces force ${}^{\mathbb{B}}\boldsymbol{F}_P$. Because the propeller is a rotating object it also generate a torque ${}^{\mathbb{B}}\boldsymbol{\tau}_P$. The purpose of the project is to fly autonomous with a constant velocity, thus ${}^{\mathbb{B}}\boldsymbol{\tau}_P$ is expected to be constant during autonomous flight. However small changes will occur, but the controller will compensate for the constant torque generated by the propeller. The Blade Element Theory calculates ${}^{\mathbb{B}}\boldsymbol{F}_P$ and ${}^{\mathbb{B}}\boldsymbol{\tau}_P$ for a propeller by summing up forces and torques for small sections of the blade, see figure 5.26.

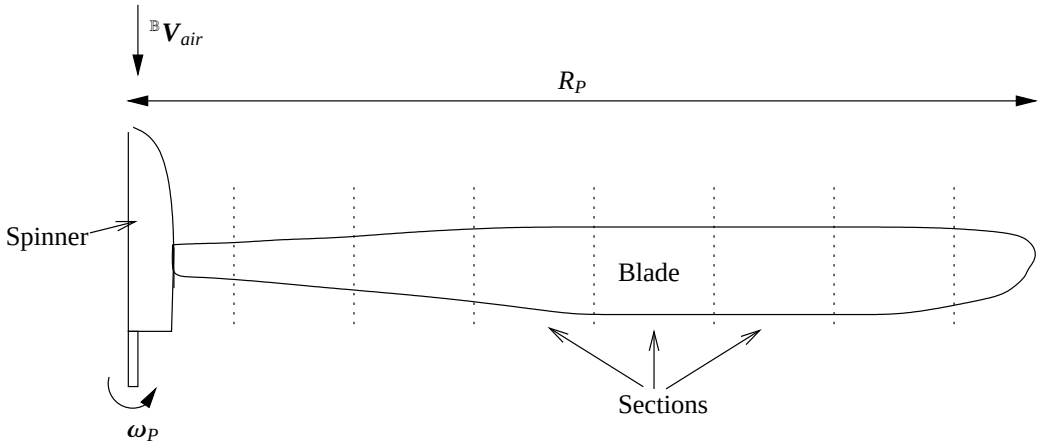


Figure 5.26: *A blade on the propeller divided in sections with the propeller radius R_P.*

Two parameters are important for the generation of thrust, which are the angular velocity of the propeller ${}^{\mathbb{B}}\boldsymbol{\omega}_P$ and the velocity of the airflow ${}^{\mathbb{B}}\boldsymbol{V}_{air}$ through the propeller. ${}^{\mathbb{B}}\boldsymbol{V}_{air}$ is perpendicular to the propeller see figure 5.27. ${}^{\mathbb{B}}\boldsymbol{V}_{air}$ is also ${}^{\mathbb{B}}V_{\text{inf},x}$, or $-{}^{\mathbb{B}}v_x$, see (5.29).

$$ {}^{\mathbb{B}}\boldsymbol{V}_{air} = \begin{bmatrix} -{}^{\mathbb{B}}v_{\mathbb{B},x} \\ 0 \\ 0 \end{bmatrix} \tag{5.29} $$

Figure 5.28 shows a sectional view of the blade, seen from the blade tip, including the aerodynamic forces and velocities of the propeller. Each section of the blade is seen as an airfoil. BET assumes that the propeller does not rotate, it is the air which is moving, so the airflow $\boldsymbol{V}_P$ has

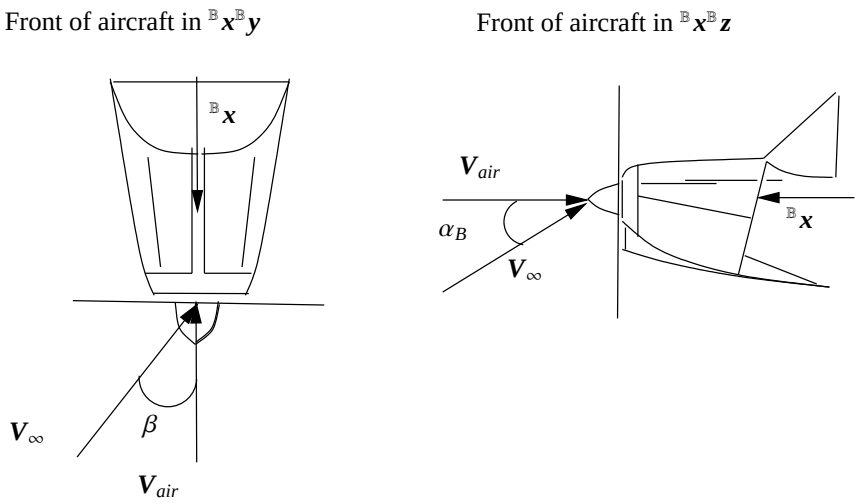


Figure 5.27: *The figure show the how $\boldsymbol{V}_\infty$ is projected onto $\boldsymbol{V}_a$ in the $^{\mathbb{B}}\boldsymbol{x}^{\mathbb{B}}\boldsymbol{y}$-plane and the $^{\mathbb{B}}\boldsymbol{x}^{\mathbb{B}}\boldsymbol{z}$-plane.*

two components: $\boldsymbol{V}_{air}$, and $\boldsymbol{V}_\omega$ the velocity of the airflow coursed by the rotation of the blade, see (5.30).

$$\boldsymbol{V}_P = \boldsymbol{V}_{air} + \boldsymbol{V}_\omega \tag{5.30}$$

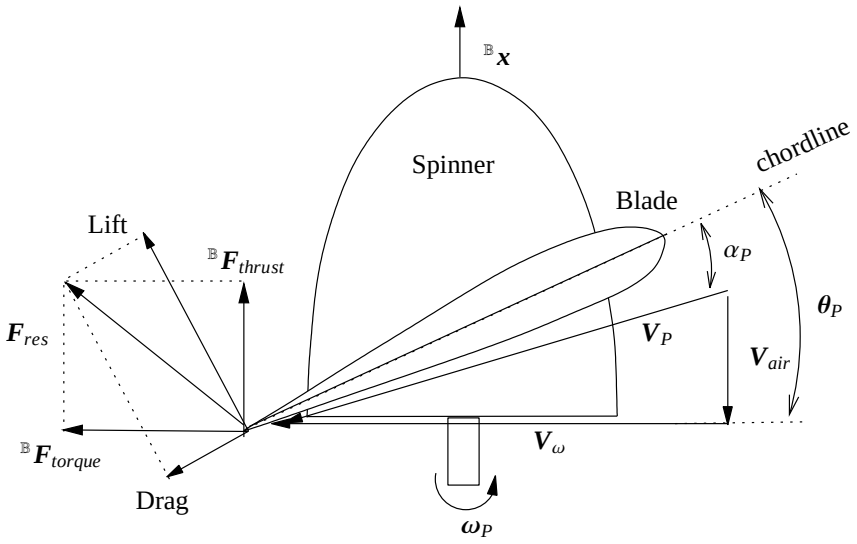


Figure 5.28: *A section of the blade seen from the blade tip, toward the spinner of the propeller. The figure show: the velocity components, the angular velocity, the aerodynamic forces: Lift, and Drag, and Thrust.*

The generation of thrust depends on $\boldsymbol{V}_P$, angle of attack α_P, and the shape on the airfoil. θ_P is the pitch angle of the blade section.

When the propeller is rotating two aerodynamic forces are generated. Lift is perpendicular to the chord line of the airfoil, and drag is parallel to the chord line. These forces generating the resulting force $\boldsymbol{F}_{res}$. Thrust ${}^{\mathbb{B}}\boldsymbol{F}_{thrust}$ is defined as the projection of ${}^{\mathbb{B}}\boldsymbol{F}_{res}$ onto ${}^{\mathbb{B}}\boldsymbol{x}$ The other component of $\boldsymbol{F}_{res}$ is ${}^{\mathbb{B}}\boldsymbol{F}_{torque}$ which is generating a torque around CoG, and this force is neglected.

5.2.2 Analysis of the Propeller

The propeller is analyzed using the program JavaProp. The purpose of the analysis is to determine the coefficients for the function

$$ {}^{\mathbb{B}}\boldsymbol{F}_P = f(\boldsymbol{V}_{air}, \boldsymbol{\omega}_P) \tag{5.31} $$

In order to find this function, JavaProp is used to find values of ${}^{\mathbb{B}}\boldsymbol{F}_P$ to different $\boldsymbol{V}_{air}$ and $\boldsymbol{\omega}_P$. This gives a surface in 3D and $f(\boldsymbol{V}_{air}, \boldsymbol{\omega}_P)$ is fitted to this function using a least square solution [1]. Two steps are needed in order to perform the analysis: determination of the geometric properties of the propeller, and choosing a working area for ${}^{\mathbb{B}}\boldsymbol{\omega}_P$ of the propeller.

Geometric properties This includes: diameter of the propeller, number of blades, diameter of spinner, and properties for each section of the blade: chord line, pitch angle θ_P, and airfoils shape.

Working area for $\boldsymbol{\omega}_P$ is min $2000\frac{2\pi}{min}$, which is the recommend minimum angular velocity for the motor. The maximal value is $8500\frac{2\pi}{min}$, the maximal angular rotation of the propeller.

The result of the analysis is shown on figure 5.29.

Figure 5.29 shows ${}^{\mathbb{B}}\boldsymbol{F}_P$ when $\boldsymbol{V}_{air}$ and $\boldsymbol{\omega}_P$ vary. Generally, ${}^{\mathbb{B}}\boldsymbol{F}_P$ is increasing when $\boldsymbol{\omega}_P$ is increasing, and it is decreasing when $\boldsymbol{V}_{air}$ increases. When $\boldsymbol{V}_{air}$ is zero, a loss of thrust is observed. This is the static thrust, where no airflow exists and the propeller must create its own airflow. This causes the efficiency of the propeller to decrease.

BET is not precis when the power load is high. These high power loads occur for this type of propeller at low $\boldsymbol{V}_{air}$, so BET is not precise in the this region, which for $\boldsymbol{\omega}_P = 2000\frac{2\pi}{\text{min}}$ is between $0\frac{\text{m}}{\text{s}}$ and $5\frac{\text{m}}{\text{s}}$. This region increases when $\boldsymbol{\omega}_P$ increase, and is for $\boldsymbol{\omega}_P = 8500\frac{2\pi}{\text{min}}$ between $0\frac{m}{s}$ and $20\frac{m}{s}$. The purpose of the project is to control the aircraft, when it is flying in the air with a constant ${}^{\mathbb{B}}\boldsymbol{\omega_P}$. Under these condition would ${}^{\mathbb{B}}\boldsymbol{V}_{air}$ be in area where BET is precise.

$\boldsymbol{V}_{air}$ must not be the region, where BET fails. It is therefore chosen to use the area shown on figure 5.30.

(b) on figure 5.29 is the least square fit of the data in (a). The data is fitted with a polynomial of the type:

$$ {}^{\mathbb{B}}\boldsymbol{F}_P(\boldsymbol{V}_{air}, \boldsymbol{\omega}_P) = a\boldsymbol{V}_{air} + b\boldsymbol{V}_{air}^2 + c\boldsymbol{\omega}_P + d\boldsymbol{\omega}_P^2 + e \tag{5.32} $$

Where the coefficients are found to be

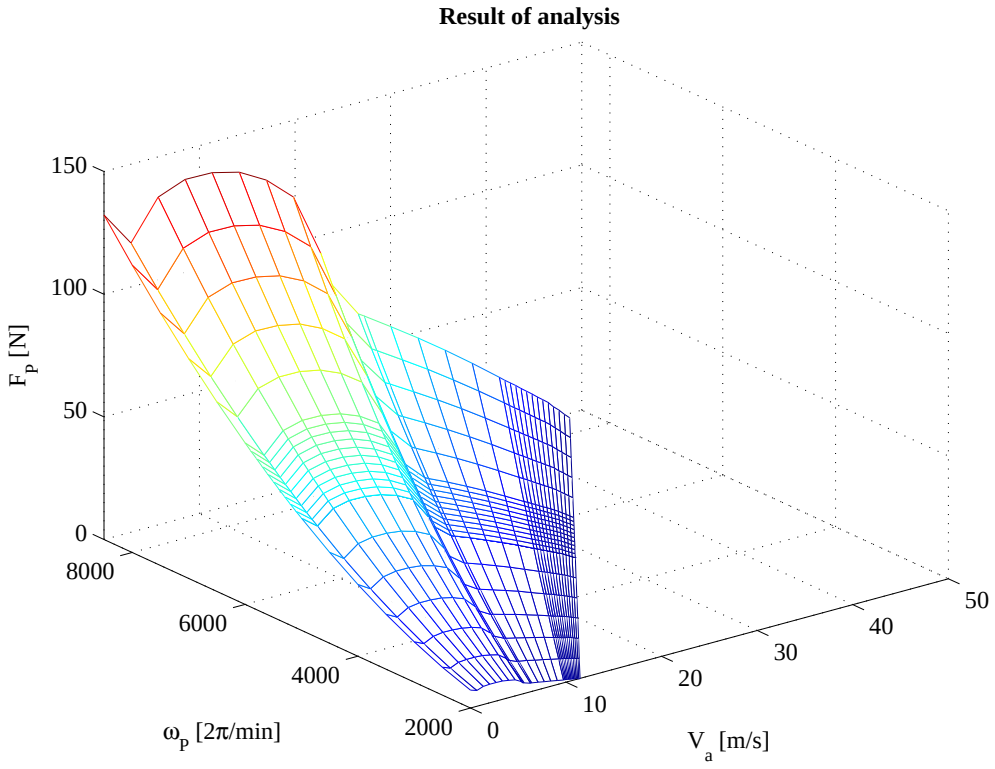


Figure 5.29: ${}^{\mathbb{B}}\boldsymbol{F}_P$ *is shown as a function of* $\boldsymbol{\omega}_P$ *and* $\boldsymbol{V}_{air}$

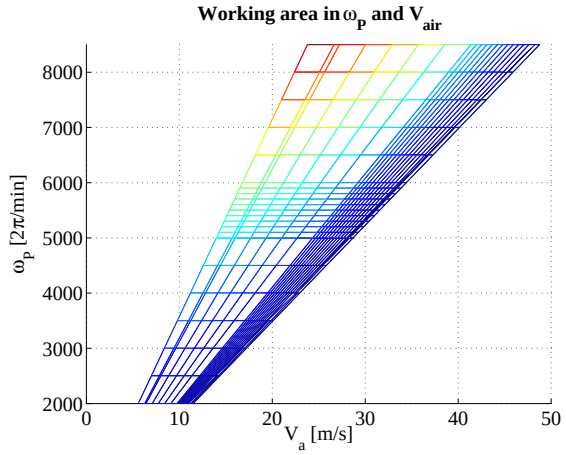
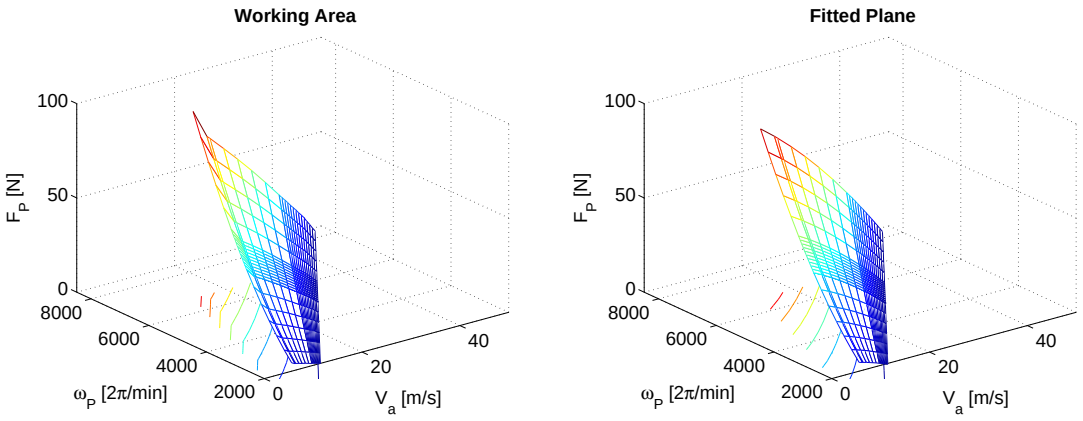


Figure 5.30: *The graph shows the working area in the* ${}^{\mathbb{B}}\omega_P{}^{\mathbb{B}}\boldsymbol{V}_{air}$*-plane.*



(a) (b)

Figure 5.31: *(a) show the region of the analysis of the propeller which is chosen. (b) show the Least Square solution to a 3 dimensional polynomial*

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} -0.2550 \\ -0.0363 \\ 0.0012 \\ 1.1935 \cdot 10^{-6} \\ 0.0145 \end{bmatrix} \tag{5.33}$$

The error between the analysis and fitted polynomial is shown on figure 5.32.

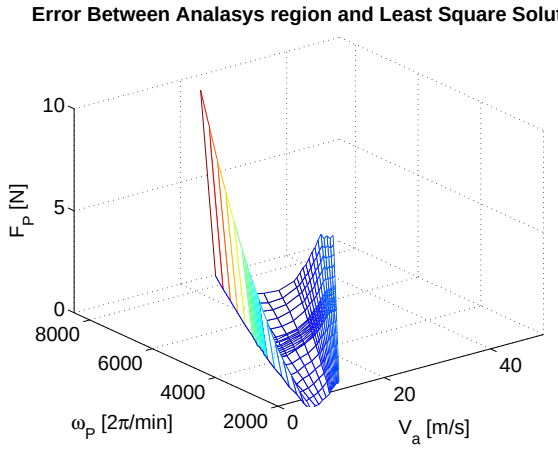


Figure 5.32: *The graph shows the error between the results of the analysis and the least squarer solution.*

5.3 Aerodynamic Model

This section contains the transformation from aerodynamic coefficients to actual forces and torques at CoG and around the axes of $\mathbb{B}$. First, it is described how the forces and torques are calculated from the coefficients for the aerodynamic forces and torques. The forces are then rotated from $\mathbb{A}$ to $\mathbb{B}$, and adding the aerodynamic forces contribution is added to the aerodynamic torque.

Calculation of Forces and Torques

To find the aerodynamic forces on the aircraft, we use the generel equations for: lift, drag, and side-force reveiling (5.34). And the corresponding equations of aerodynamic torque [3] giving (5.35).

$$^{\mathbb{B}}\boldsymbol{F}_A = {}^{\mathbb{B}}_{\mathbb{A}}\boldsymbol{q} \otimes \begin{bmatrix} \bar{q}S\, C_{^{\mathbb{A}}\boldsymbol{F}_A} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{A}}\boldsymbol{q}^* \tag{5.34}$$

$$^{\mathbb{B}}\boldsymbol{\tau}_A = \bar{q}S \begin{bmatrix} b & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & b \end{bmatrix} C_{^{\mathbb{B}}\boldsymbol{\tau}_A} \tag{5.35}$$

Where $S = 0.795\text{m}^2$ is the reference area of the aircraft. $b = 2.405\text{m}$ is the reference wing span. $c = 0.32\text{m}$ is the mean chord lenght. And $\bar{q} = \frac{\rho V_\infty^2}{2}$ is the dynamic pressure, where: ρ is the density of the air, and V_∞ is the relative wind speed [3].

The Aerodynamic Model

To find the force $^{\mathbb{B}}\boldsymbol{F}_{aero}$ and the torque $^{\mathbb{B}}\boldsymbol{\tau}_{aero}$, we will combine the propeller and aircraft aerodynamics. Since the aircraft aerodynamic force $^{\mathbb{B}}\boldsymbol{F}_A$ works in the Center of Aerodynamics, it has an arm down to the CoG, taking this into account the following is an equation for the resulting forces (5.36) and torques (5.37), where $\times$ is the vector cross product.

$$^{\mathbb{B}}\boldsymbol{F}_{aero} = {}^{\mathbb{B}}_{\mathbb{A}}\boldsymbol{q} \otimes \begin{bmatrix} \bar{q}S\, C_{^{\mathbb{A}}\boldsymbol{F}_A} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{A}}\boldsymbol{q}^* + {}^{\mathbb{B}}\boldsymbol{F}_P \tag{5.36}$$

$$^{\mathbb{B}}\boldsymbol{\tau}_{aero} = \bar{q}S \begin{bmatrix} b & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & b \end{bmatrix} C_{^{\mathbb{B}}\boldsymbol{\tau}_A} + \boldsymbol{R}_{CoA} \times {}^{\mathbb{B}}\boldsymbol{F}_A \tag{5.37}$$

Chapter 6

Aircraft Dynamics and Kinematics

This chapter describes the dynamics and kinematics of the aircraft, when it is affected by forces and torques. The dynamical model is splitted in two models. One model describes translatorial accelerations and the other model describes angular accelerations. The kinematic model describe the angle of the aircraft with respect to $\mathbb{R}$. The last section of the chapter is a validation of the models.

The dynamics and kinematics of the aircraft are splitted in translatorial and angular accelerations and velocities, see figure 6.1.

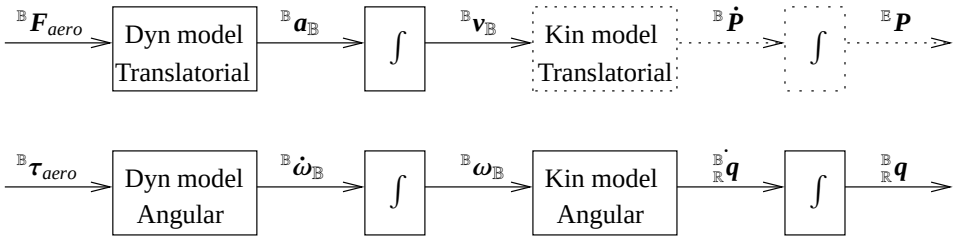


Figure 6.1: *Inputs and outputs are shown for the: translatorial dynamical model, the angular dynamical model, the translatorial kinematic model and angular kinematic model.*

${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ is the nomenclature for $\boldsymbol{\omega}_{\mathbb{B}}$ the angular velocity of $\mathbb{B}$ seen from $\mathbb{B}$, and measured with respect to the inertial coordinate system $\mathbb{R}$. This notation is used for both acceleration and velocity.

Translatorial dynamical model calculates the translatorial acceleration along ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$ when a force is applied to CoG.

Translatorial kinematic model calculates the time derivative of the position ${}^{\mathbb{E}}\dot{\boldsymbol{P}}$. The linear kinematic model is used to describe the position of CoG with respect to $\mathbb{E}$. The position of the aircraft is needed if it should follow a path, but this is not the purpose with the aircraft. This is out of the scope of the project, thus the translatorial kinematic models isn't developed.

Angular dynamical model calculates the angular acceleration of the aircraft around ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$. The input to the model is torques acting around ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$ and ${}^{\mathbb{B}}\boldsymbol{z}$.

Angular kinematic model determines the orientation of $\mathbb{B}$ with respect to $\mathbb{R}$. This orientation is described using quaternions. The input for the kinematic model is ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. The output ${}^{\mathbb{B}}_{\mathbb{R}}\dot{\boldsymbol{q}}$ is integrated so the result is the quaternion ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$.

6.1 Dynamical Model

The dynamical model, is divided in two parts the translatorial dynamics described in section 6.1.1, and the angular dynamics described in section 6.1.2.

6.1.1 Translatorial dynamics

The resulting force on the aircraft ${}^{\mathbb{B}}\boldsymbol{F}_{res}$ is the sum of the external forces ${}^{\mathbb{B}}\boldsymbol{F}_{aero}$ and ${}^{\mathbb{B}}\boldsymbol{F}_{G}$. ${}^{\mathbb{B}}\boldsymbol{F}_{aero}$ is a sum the aerodynamics forces lift, drag, sideforce, and thrust, see chapter 5. ${}^{\mathbb{B}}\boldsymbol{F}_{G}$ is gravity acting on the aircraft.

$$ {}^{\mathbb{B}}\boldsymbol{F}_{res} = {}^{\mathbb{B}}\boldsymbol{F}_{aero} + {}^{\mathbb{B}}\boldsymbol{F}_{G} \tag{6.1} $$

The gravity force is acting towards the center of the earth, thus the force is parallel with ${}^{\mathbb{R}}\boldsymbol{z}$.

$$ {}^{\mathbb{R}}\boldsymbol{F}_{G} = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{6.2} $$

All forces in (6.1) is given in $\mathbb{B}$, so it is necessary to rotate ${}^{\mathbb{R}}\boldsymbol{F}_{G}$ to $\mathbb{B}$. This rotation is done by a quaternion multiplication[1].

$$ {}^{\mathbb{B}}\boldsymbol{F}_{res} = {}^{\mathbb{B}}\boldsymbol{F}_{aero} + {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q} \otimes \begin{bmatrix} {}^{\mathbb{R}}\boldsymbol{F}_{G} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}^{*} \tag{6.3} $$

(6.3) is rewritten such that it express the acceleration of the aircraft.

$$ \begin{aligned} m\,{}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}} &= {}^{\mathbb{B}}\boldsymbol{F}_{aero} + m\,{}^{\mathbb{B}}\boldsymbol{g} \\ &\Downarrow \\ {}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}} &= \frac{1}{m}{}^{\mathbb{B}}\boldsymbol{F}_{aero} + {}^{\mathbb{B}}\boldsymbol{g} \end{aligned} \tag{6.4} $$

Where ${}^{\mathbb{B}}\boldsymbol{g}$ is defined as ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ 0 \\ g \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}^{*}$

[1] The multiplication between to quaternions is given in appendix C

6.1.2 Angular Dynamics

The aerodynamic forces are not generated in CoG. The forces are created on the surface of the aircraft. Since the aerodynamic forces are not acting directly on CoG, a torque ${}^{\mathbb{B}}\boldsymbol{\tau}_{aero}$ is generated around ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$, and ${}^{\mathbb{B}}\boldsymbol{z}$ causing an angular acceleration ${}^{\mathbb{B}}\dot{\boldsymbol{\omega}}_{\mathbb{B}}$ of the aircraft.

The angular dynamical model is described by Euler's Rotational Equations of Motion[15](6.5).

$$ {}^{\mathbb{B}}\boldsymbol{\tau}_{aero} = \boldsymbol{I}\,{}^{\mathbb{B}}\dot{\boldsymbol{\omega}}_{\mathbb{B}} + {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \times (\boldsymbol{I} \cdot {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}) \tag{6.5} $$

(6.5) calculates the torque on the aircraft in $\mathbb{B}$ when an angular acceleration ${}^{\mathbb{B}}\dot{\boldsymbol{\omega}}_{\mathbb{B}}$ is applied to the aircraft. ${}^{\mathbb{B}}\boldsymbol{I}$ is the moment of inertia matrix and defined in equation (6.6).

$$ \boldsymbol{I} = \begin{bmatrix} I_y^2 + I_z^2 & -I_x I_y & -I_x I_z \\ -I_y I_x & I_x^2 + I_z^2 & -I_y I_z \\ -I_z I_x & -I_z I_y & I_x^2 + I_y^2 \end{bmatrix} \tag{6.6} $$

where I_x, I_y, and I_z are the inertia of the aircraft in ${}^{\mathbb{B}}\boldsymbol{x}$, ${}^{\mathbb{B}}\boldsymbol{y}$, and ${}^{\mathbb{B}}\boldsymbol{z}$. Since the aircraft is symmetric about the ${}^{\mathbb{B}}\boldsymbol{x}{}^{\mathbb{B}}\boldsymbol{z}$ plane, are the following products of inertia equal zero.

$$ I_x I_y = I_x I_y = I_x I_y = I_x I_y = 0 \tag{6.7} $$

(6.6) are then reduced using the assumption in (6.7), reveling (6.8).

$$ \boldsymbol{I} = \begin{bmatrix} I_y^2 + I_z^2 & 0 & -I_x I_z \\ 0 & I_x^2 + I_z^2 & 0 \\ -I_z I_x & 0 & I_x^2 + I_y^2 \end{bmatrix} \tag{6.8} $$

The inertia of the aircraft is taken from a real Cessna and scaled to the model aircraft, [10]. This results in the matrix of inertia in (6.9).

$$ \boldsymbol{I} = \begin{bmatrix} 13.4743 & 0 & 0 \\ 0 & 19.1363 & 0 \\ 0 & 0 & 27.9654 \end{bmatrix} \tag{6.9} $$

These values are not the correct inertia for the aircraft model, because the model have another size, and built of different materials. Hence it can only be used to model validation. Experiments are necessary to find the correct $\boldsymbol{I}$. The inertia can be identified with data from the test flight, or a pendulum test. The chosen matrix of inertia is used, until it is possible to find the right one.

The centrifugal term ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \times (\boldsymbol{I} \cdot {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}})$ in (6.5) describes how a velocity in one axis gives a velocity in the two other axes. This is referred to as nutation.

The purpose with the angular dynamical model is to calculate an angular acceleration, where ${}^{\mathbb{B}}\boldsymbol{\tau}_{aero}$ is an input, thus (6.5) is rewritten to (6.10):

$$ {}^{\mathbb{B}}\dot{\boldsymbol{\omega}}_{\mathbb{B}} = \boldsymbol{I}^{-1}\left({}^{\mathbb{B}}\boldsymbol{\tau}_{\boldsymbol{aero}} - {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \times (\boldsymbol{I} \cdot {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}})\right) \tag{6.10} $$

6.2 Kinematic Model

The angular kinematic model determines the orientation of the aircraft frame $\mathbb{B}$ with respect to $\mathbb{R}$. The orientation is described using quaternions,and the input to the angular kinematic model is ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. Thus the model describes, how the quaternion for the aircraft are calculated. An expression for the time derivative of the quaternion is found in appendix C on page 97. (6.11) and (6.12) describes how the time derivative ${}^{\mathbb{B}}_{\mathbb{R}}\dot{\boldsymbol{q}}$ is calculated from ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. The output is integrated and the output from the angular kinematic model is ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$.

$$ {}^{\mathbb{B}}_{\mathbb{R}}\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} \boldsymbol{S}({}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}) & {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \\ {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}^{\mathrm{T}} & 0 \end{bmatrix} \tag{6.11} $$

$$ \boldsymbol{S}({}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}) = \begin{bmatrix} 0 & -{}^{\mathbb{B}}\omega_{\mathbb{B},z} & {}^{\mathbb{B}}\omega_{\mathbb{B},y} \\ {}^{\mathbb{B}}\omega_{\mathbb{B},z} & 0 & -{}^{\mathbb{B}}\omega_{\mathbb{B},x} \\ -{}^{\mathbb{B}}\omega_{\mathbb{B},y} & {}^{\mathbb{B}}\omega_{\mathbb{B},x} & 0 \end{bmatrix} \tag{6.12} $$

6.3 Validation

The dynamic and kinematic models derived in this chapter are implemented in MatLab S-functions. This implementation has been compared to another implementation made in SimuLink SimMechanics. This test show that the implementation of the models are correct. However It do not verify that the models are correct, but they are likely to be so.

SimMechanics is a toolbox, where a mechanical system are build of different premade blocks. The main property of this toolbox are that the kinematics and dynamics of the mechanical system is contained in these blocks. For the aircraft, it is only necessary to give: the mass, and the moment of inertia matrix, when the system is built. Thus it is not necessary to write the equations for the dynamic and kinematic of the aircraft. Therefore are this toolbox used to validate the equations.

The validation was been conducted as shown in figure 6.2. The same inputs are given to both implementations. The output from both implementations is measured and compared.

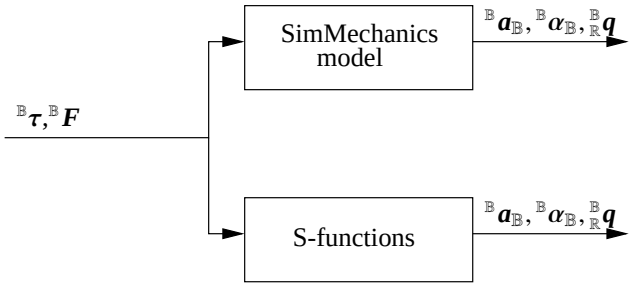


Figure 6.2: *The models take forces and torques as input and give: translatorial acceleration, angular acceleration, and the quaternion of the aircraft with respect to* $\mathbb{R}$.

The result of this test shows that the two implementations give the same result, so the imple-

mentation of the S-function is correct with respect to the SimMechanics model. Tests have shown that the SimMechanics model is a more precise implementation than the S-functions, so the SimMechanics model is used in the implementation.

Chapter 7

Control of Aircraft

This chapter contains linearization and control of the aircraft. The aircraft is linearized using a MIMO nonlinear control feedback technique. The linearized system is controlled by a cascading controller, with an inner control loop, which stabilizes ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. An outer quaternion control loop, controls the orientation of the aircraft. Following after the design, we investigate the noise and model perturbation effects on the controller.

The aircraft is described by three models, the aerodynamic model, the dynamic model, and the kinematic model. All these models are nonlineare, so in order design a controller it is necessary to make a linearization. Figure 7.1 show the control design of the aircraft.

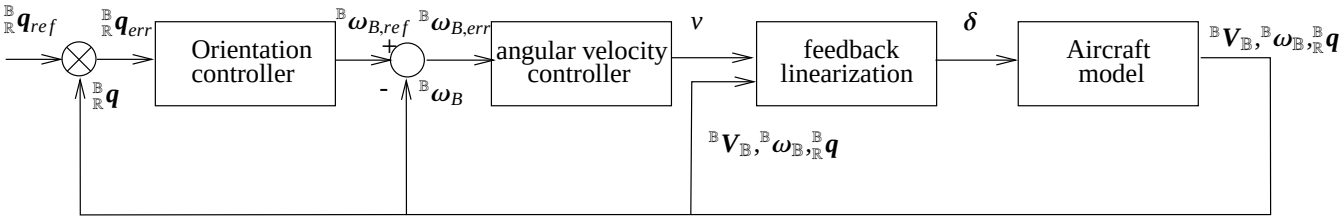


Figure 7.1: *The figure show the control structure of the aircraft. There is a quaternion controller to control the orientation of the aircraft, an angular velocity to stabilize ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, and a feedback linearization to linearize the nonlinear aircraft model.*

The orientation controller controls the orientation of the aircraft ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$. The input to the controller is ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref}$, the desired orientation of the aircraft. The difference between ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref}$ and the current orientation of the aircraft ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$ is found through a quaternion multiplication. This quaternion ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{err}$ is converted to ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},ref}$, which is the reference for the angular velocity controller.

The angular velocity controller stabilize ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ with a state-space controller. The input is ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},err}$ which are ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},ref}$ minus ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. The output v is feed into the feedback linearization.

Feedback linearization for a MIMO system is used to linearized the aircraft model. The input for this model are the control signal v, and all the states of the aircraft model, ${}^{\mathbb{B}}\boldsymbol{V}_{\mathbb{B}}$, ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$,

and ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$. The output from the feedback linearization is the deflection angles on flaps δ_f, aileron δ_a, elevator δ_e, and rudder δ_r.

The Aircraft model include the aerodynamic model, the dynamic model and the kinematic model. The inputs are deflection angles and the outputs are the states of the system.

7.1 Feedback Linearization

The three models representing the aircraft are all nonlinear models, and it is necessary to make a linearization, and feedback linearization is chosen. The purpose is to transform the nonlinear system into a linear one, instead of linearizing it around a working point as it is done in Taylor linearization. Instead of making a linearization, which work near a working point, we try to make a controller which linearized the model in all working points. The linearization is done using the methods given in [7] and [6, chapter 5]

The aircraft model is written on the nonlinear form:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{7.1}$$

Where the states are

$$x = \begin{bmatrix} {}^{\mathbb{B}}\boldsymbol{V}_{\mathbb{B}} \\ {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \\ {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q} \end{bmatrix} \tag{7.2}$$

There are five possible inputs, the four deflection angle, δ_f, δ_a, δ_e, and δ_r, and the angular velocity of the propeller ${}^{\mathbb{B}}\boldsymbol{\omega}_P$. The goal for this project is to control ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$ and this is done by using δ_a, δ_e, and δ_r, since they control roll, pitch and yaw. δ_f and ${}^{\mathbb{B}}\boldsymbol{\omega}_P$ control ${}^{\mathbb{B}}\boldsymbol{v}_{\mathbb{B}}$ and this state is not controlled, because it is out of the scope for the project, so they are treated as constants, where δ_f is zero and ${}^{\mathbb{B}}\boldsymbol{\omega}_P$ is a constant in the working area for the propeller specified in section 5.2.2 on page 45. The input vector is defined as:

$$u = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \tag{7.3}$$

$f(x)$ and $g(x)$ are found by inserting all the equation necessary to describe the entire aircraft model. The equations are seen in aircraftmodel.m, located on the CD in matlab/controller/. $g(x)$ contains the terms which are dependent on u. The rest of the terms are independent of u, and are placed in $f(x)$. $h(x)$ is the output function representing the controlled output $y = {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, because the purpose is to stabilize ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$.

$$h(x) = {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \tag{7.4}$$

The purpose with feedback linearization is to convert the system given in (7.1) into the linear system:

$$\begin{aligned} \dot{x} &= Fx + Gu \\ y &= Hx \end{aligned} \tag{7.5}$$

This is done by rewriting (7.1) as:

$$\begin{aligned} \dot{x} &= Fx + G\gamma(x)(u - \alpha(x)) \\ y &= h(x) \end{aligned} \tag{7.6}$$

The functions $\gamma(x)$ and $\alpha(x)$[1] represents the nonlinearities in the system. (7.7) is used as the linearization control law for the system in (7.6). The function $\beta(x)$ is equal $\gamma^{-1}(x)$. v is the new input to the system.

$$u = \alpha(x) + \beta(x)v \tag{7.7}$$

Equation (7.7) is inserted in (7.6) and gives the system in (7.8). Figure 7.2 show the principle of feedback linearization. Equation (7.7) cancels the nonlinearities, such that the system appears as a linear system.

$$\begin{aligned} \dot{x} &= Fx + G(x)v \\ y &= h(x) \end{aligned} \tag{7.8}$$

The next section describes how $\alpha(x)$ and $\beta(x)$ are found. In order to do this it is necessary to find the relative degree of the system.

7.1.1 Relative Degree

The relative degree r is used to find $\alpha(x)$ and $\beta(x)$. The relative degree describes whether the output is independent of the input. If the relative degree is zero, it is not possible to control the output. The relative degree is found using the Lie Derivative. The used Lie Derivative nomenclature is defined in (7.9), see [7, p.512] :

$$\begin{aligned} L_f h(x) &= \frac{\partial h}{\partial x} f(x) \\ L_g L_f h(x) &= \frac{\partial L_f h(x)}{\partial x} g(x) \\ L_f^k h(x) &= \frac{\partial L_f^{k-1} h(x)}{\partial x} f(x) \\ L_f^0 h(x) &= h(x) \end{aligned} \tag{7.9}$$

The conditions in (7.10) are used to find the relative degree of a system.

$$L_g L_f^{i-1} h(x) = 0, \qquad i = 1, 2, ..., r-1; \qquad L_g L_f^{r-1} h(x) \neq 0 \tag{7.10}$$

[1] $\alpha(x)$ and $\beta(x)$ are standard nonlinear control theory functions used to represent nonlinearities. They have in this contents nothing to do with angle of attack and side slip.

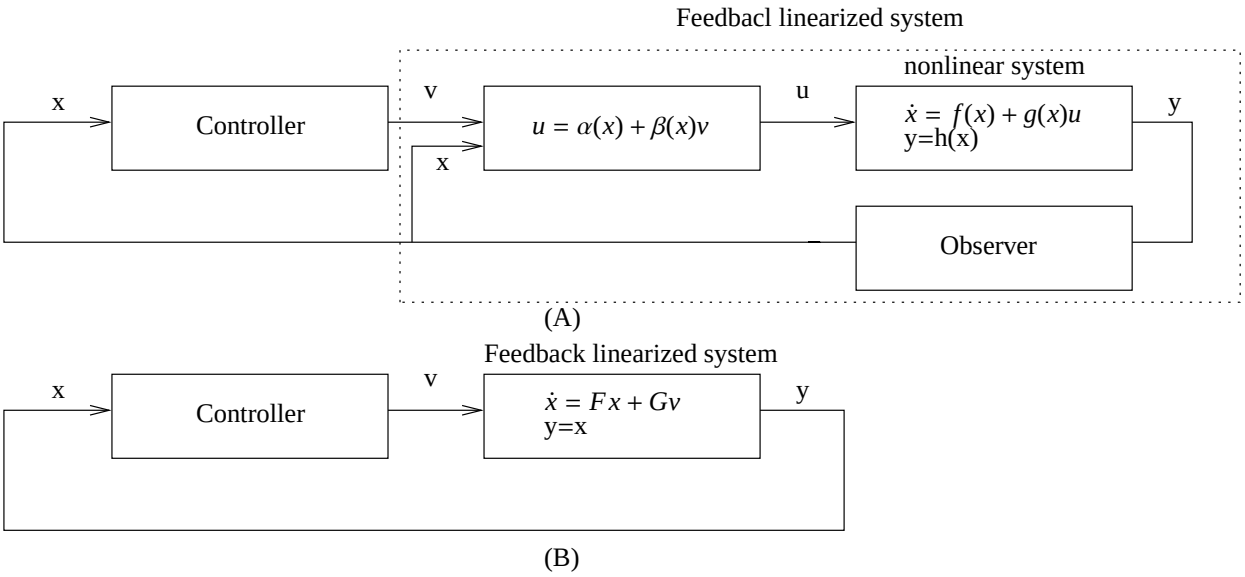


Figure 7.2: *The figure shows the principle of feedback linearization. Figure A shows the actual system, while figure B shows how the system is seen from the controllers point of view.*

Equation (7.10) describes how the relative degree is found for a SISO system, but the aircraft model is a MIMO system, and can be written as a MIMO system having ten states, three input, and three output.

$$\begin{aligned} \dot{x} &= f(x) + \begin{bmatrix} g_1(x) & g_2(x) & g_3(x) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \\ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} &= \begin{bmatrix} h_1(x) \\ h_2(x) \\ h_3(x) \end{bmatrix} \end{aligned} \tag{7.11}$$

The system in (7.11) has three inputs, and therefore is the relative degree vector given as $r = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$ and the relative degree is found using the properties in (7.12), see [6, p. 220].

$$\begin{aligned} &L_{g,j} L_f^k h_i(x) = 0, \qquad k = 1, 2, \cdots, r-1 \quad , 1 \leq j \leq m, \quad 1 \leq i \leq m \\ &L_{g,j} L_f^{r_i - 1} h_i(x) \neq 0 \end{aligned} \tag{7.12}$$

where $m = 3$, the number of input for the aircraft model. This is done for the aircraft system, and the relative degree of the system is $r = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. It is therefore possible to control the outputs, which in our case are ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, with the inputs. The inputs are the deflection angles of aileron, elevator, and rudder. The calculations for finding the relative degree is done using MatLab symbolic toolbox, and can is implemented in the m-file control/matlab/relativeDegree.m on the CD.

$\alpha(x)$ and $\beta(x)$ is given as:

$$\alpha(x) = -\boldsymbol{A}^{-1}(x)\boldsymbol{b}(x) \tag{7.13}$$

$$\beta(x) = \boldsymbol{A}^{-1}(x) \tag{7.14}$$

where $\boldsymbol{A}(x)$ and $\boldsymbol{b}(x)$ is:

$$\boldsymbol{A}(x) = \begin{bmatrix} L_{g1}L_f^0 h_1(x) & L_{g2}L_f^0 h_1(x) & L_{g3}L_f^0 h_1(x) \\ L_{g1}L_f^0 h_2(x) & L_{g2}L_f^0 h_2(x) & L_{g3}L_f^0 h_2(x) \\ L_{g1}L_f^0 h_3(x) & L_{g2}L_f^0 h_3(x) & L_{g3}L_f^0 h_3(x) \end{bmatrix} \tag{7.15}$$

$$\boldsymbol{b}(x) = \begin{bmatrix} L_f^{r1} h_1(x) \\ L_f^{r2} h_2(x) \\ L_f^{r3} h_3(x) \end{bmatrix} \tag{7.16}$$

The expressions of $\alpha(x)$ and $\beta(x)$ are found in the matlab function nonInteractive.m. The control law $u = \alpha(x) + \beta(x)v$ is implemented in the function matlab/model/control/controllin.m. It is now possible to find the linear functions in (7.17). $\boldsymbol{F}$ is the linear terms in the aircraft model $\dot{x} = f(x) + g(x)u$, this mean terms which depends on one state. These terms is found and $\boldsymbol{F}$ is given (7.18). $\boldsymbol{F}$ is zero, because there is no linear terms in the aircraft model, and $\boldsymbol{G}$ is an identity matrix.

$$\dot{x} = Fx + G\gamma(x)(u - \alpha(x)) \tag{7.17}$$

$$\boldsymbol{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.18}$$

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.19}$$

The control law in (7.7) control the state ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, but the other states ${}^{\mathbb{B}}\boldsymbol{V}_{\mathbb{B}}$ and ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$ is not in the feedback loop. These states is called the internal dynamic also defined as η. The entire system is given as:

$$\dot{\eta} = f_0(\eta, x) \tag{7.20}$$

$$\dot{x} = Fx + G\gamma(x)(u - \alpha) \tag{7.21}$$

$$y = h(x) \tag{7.22}$$

It is important to secure that internal dynamic is stable, because these states cannot be controlled by the input. Chapter 8 on page 66 contains an stability analysis of the aircraft controlled by the control law in (7.7). The next section would test if the system is noninteractive.

7.1.2 Noninteractive Control

A MIMO system is noninteractive if the input u_i controls the output y_i, but have no influence on $y_{j\neq i}$. A system which is feedback linearized is noninteractive if the matrix $\boldsymbol{A}(0)$ is nonsingular. The states is ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. These states is set to zero, and the rank of $\boldsymbol{A}(0)$ is 3. This mean that $\boldsymbol{A}(0)$ is nonsingular and the system is noninteractive. The result is found in nonInteractive.m.

7.2 Angular Velocity Controller

After the feedback linearization is the system for the aircraft written in the form:

$$\begin{aligned} \dot{x} &= Gv \\ y &= Hx \end{aligned} \tag{7.23}$$

where the input is v and x is ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, see figure 7.3. This system is controlled with a state space controller. The aircraft is a system where it is better to have a steady state error than an overshoot. A slow controller would introduce a steady state error, but prevent the aircraft from making very fast maneuvers.

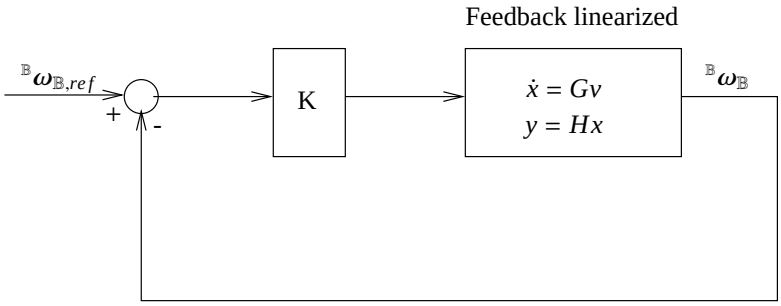


Figure 7.3: *The controller structure of the angular velocity controller. K is the state space controller.*

The state space system is controlled by the feedback law $v = -\boldsymbol{K}e$, and the closed-loop dynamic is given as $x = (\boldsymbol{F} - \boldsymbol{GK})e$, where the closed-loop poles of the system is the eigenvalues of $(\boldsymbol{F} - \boldsymbol{GK})$. Because of the structure of the system in (7.23) is the system simplified to:

$$\dot{x} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 & 0 \\ 0 & 0 & -k_3 \end{bmatrix} e \tag{7.24}$$

where k_1, k_2, and k_3 is the poles of the closed-loop system. The poles for the system is place in the left half plane of the S-domain in -10, so k_1, k_2, and k_3 is equal 10, and $e = {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},ref} - {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. The system is simulated with the chosen poles. The result can be seen in figure 7.4. The aircraft performs a 180° roll. Figure 7.4 shows the angular rates of the aircraft during the 180° roll. The simulation shows that the system follows the reference nicely. It is noted that there is a small pitch disturbance in the first 10s, due to that the simulator isn't trimmed. It is also noted that the orientation change first start after the simulator have trimmed itself (after 10s).

Figure 7.5 shows the position of the aircraft in the navigational coordinate system, during the simulation.

7.3 Orientation Controller

Quaternions are used to describe the orientation of the aircraft, thus to control the orientation, we control the quaternion. The design of the controller is shown on figure 7.6.

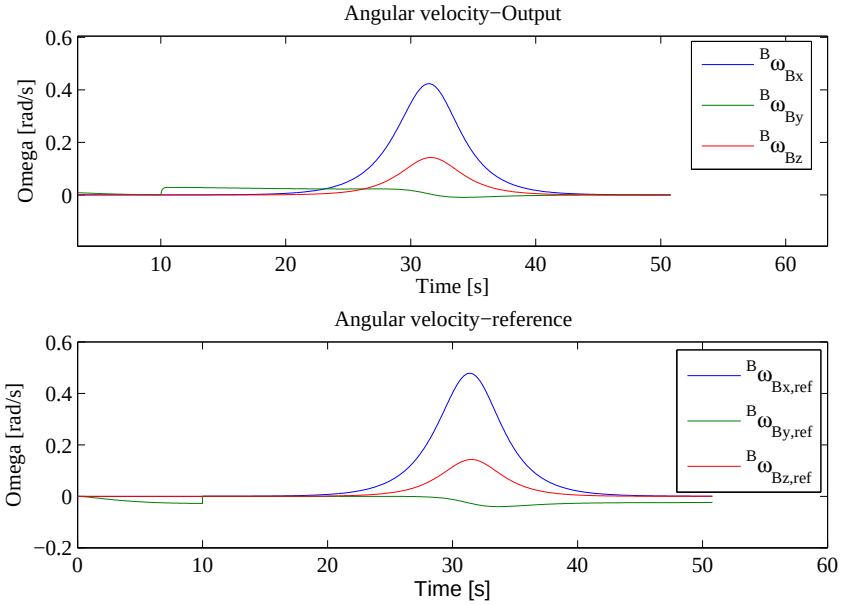


Figure 7.4: *The figure shows* ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ *of the aircraft in the top graph, the other graph is the reference* ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},ref}$.

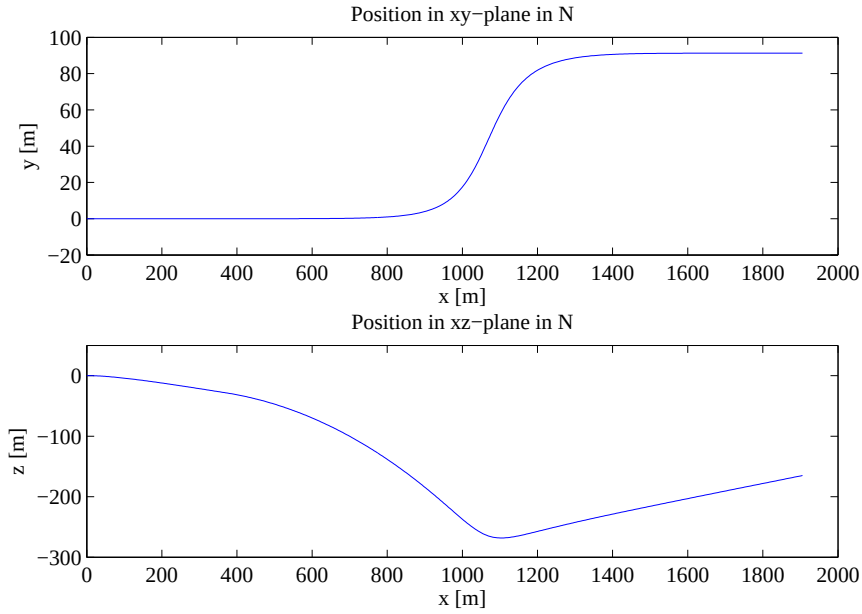


Figure 7.5: *The position of the aircraft in the* ${}^{\mathbb{N}}\boldsymbol{x}{}^{\mathbb{N}}\boldsymbol{y}$*-plane in the top graph and the* ${}^{\mathbb{N}}\boldsymbol{x}{}^{\mathbb{N}}\boldsymbol{z}$*-plane is shown in the other graph.*

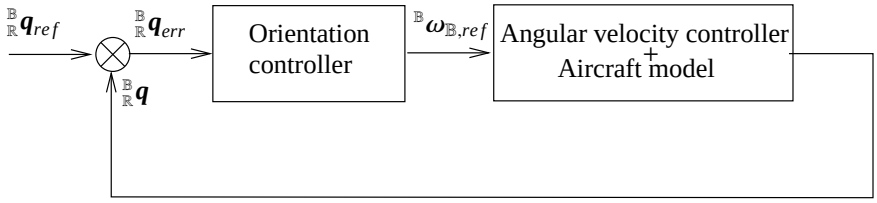


Figure 7.6: *The design of the orientation controller using quaternions.* ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref}$ *is the desired orientation for the aircraft, while* ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$ *is the orientation of the aircraft.*

The error ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{err}$ between the wanted orientation of the aircraft ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref}$ and the actual position ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$ is found using (7.25).

$$ {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{err} = k_q {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}^* \otimes {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref} \tag{7.25} $$

where ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{err}$ represent the remaining rotation between ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref}$ and ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$. With this control law gives a big error a big control signal, but the actuators can go in saturation. δ_a have saturation limits in $\pm 20°$, δ_e have saturation in $\pm 35°$, and δ_r have saturation in $\pm 25°$. The maximum rotation for the aircraft is 180°, a bigger angle courses the aircraft to rotate the other way around. A 180° rotation would give high deflection angles, and the actuators would go in saturation. k_q is the proportional gain of the controller, and this is used to tune the controller. It is possible to change make the gain so small that the actuators don't go in saturation under a maximum rotation. k_q is chosen to be 1, and this gain would give saturation in the actuators, so the following is done:

$$ \begin{aligned} {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},ref} &= k_q \begin{bmatrix} {}^{\mathbb{B}}_{\mathbb{R}}q_{1,err} \\ {}^{\mathbb{B}}_{\mathbb{R}}q_{2,err} \\ {}^{\mathbb{B}}_{\mathbb{R}}q_{3,err} \end{bmatrix} {}^{\mathbb{B}}_{\mathbb{R}}q_{4,err} \\ &= k_q \hat{\boldsymbol{e}} \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\theta}{2}\right) \\ &= k_q \hat{\boldsymbol{e}} \frac{1}{2} \sin(\theta) \end{aligned} \tag{7.26} $$

The result of (7.26), is a control signal, which is given as the function $\frac{1}{2}\sin(\theta)$. The control signal is small if the error is near $\theta = 0$ or $\theta = \pi$, and it will be at a maximum at $\theta = \frac{pi}{2}$. With this solution will the actuators not go in saturation so often, and it prevent the aircraft from making very fast maneuvers, if it must rotate with an angle near π.

Unfortunately, the control law in (7.26) is singular when ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{4,err} = 0$. To over come this, we add a term, which decide the rotation direction, when a 180° is wanted.

$$ {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},ref} = k_q \begin{bmatrix} {}^{\mathbb{B}}_{\mathbb{R}}q_{1,err} \\ {}^{\mathbb{B}}_{\mathbb{R}}q_{2,err} \\ {}^{\mathbb{B}}_{\mathbb{R}}q_{3,err} \end{bmatrix} ({}^{\mathbb{B}}_{\mathbb{R}}q_{4,err} + 0.0001 \cdot sign({}^{\mathbb{B}}_{\mathbb{R}}q_{4,err})) \tag{7.27} $$

This controller is implemented in the S-function qcontroller.m. The controller is tested on the system. Figure 7.7 shows the quaternion for a 180° rotation in yaw. The aircraft start with the quaternion ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$, flying along ${}^{\mathbb{B}}\boldsymbol{x}$. After 10sec the quaternion is changed

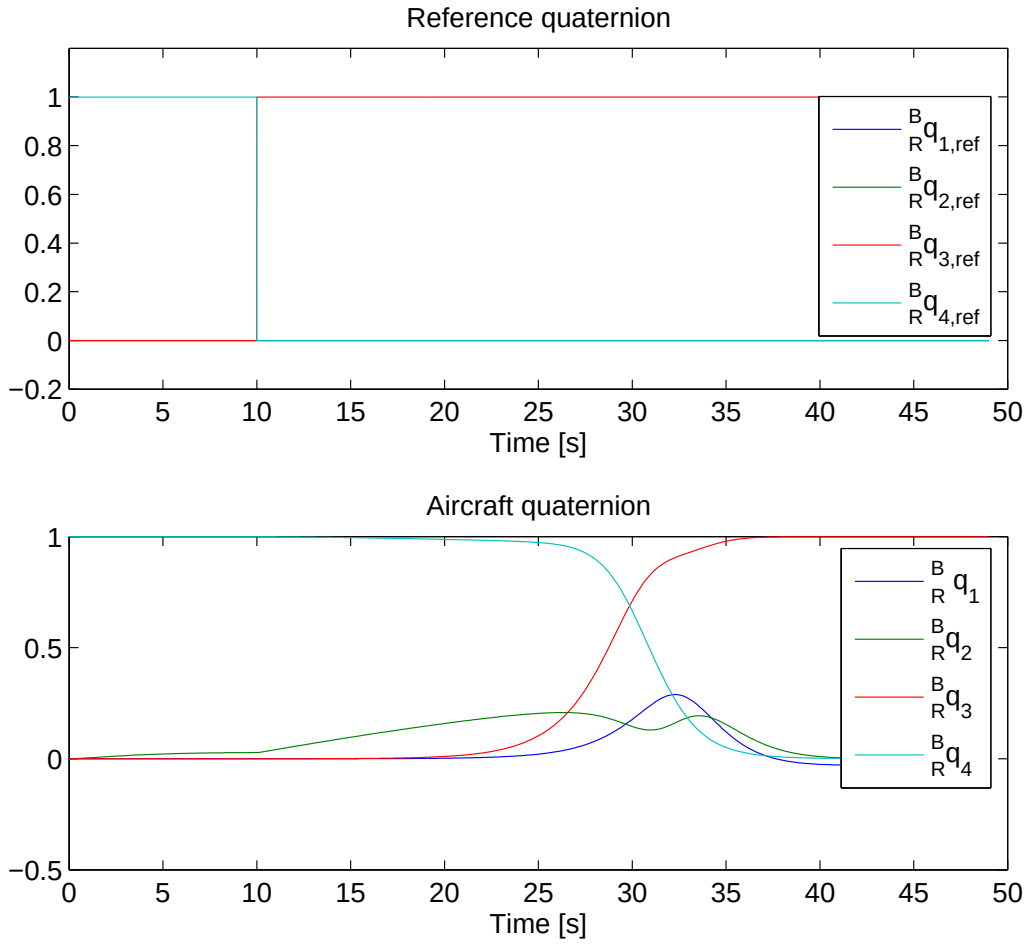


Figure 7.7: *A 180° rotation is made in yaw. The top graph is* $^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref}$*, and the bottom graph is* $^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$.

to ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}_{ref} = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}^{\mathrm{T}}$, a 180° rotation around yaw. Figure 7.7 shows this rotation. The reference is given at 10sec, and it is seen that the aircraft rotate slowly at the start. The rotation is finish after 35sec. It can be seen there is a small steady state error at the end of the rotation.

Figure 7.8 shows the position of the aircraft in $\mathbb{R}$. The top graph shows the aircraft turn 180° in the ${}^{\mathbb{N}}\boldsymbol{x}^{\mathbb{N}}\boldsymbol{y}$. The other graph show the ${}^{\mathbb{N}}\boldsymbol{x}^{\mathbb{N}}\boldsymbol{z}$-plane, where it can be seen how the aircraft takes off. Under the maneuverer is the aircraft losing height. When it finishes the rotation it starts climbing again.

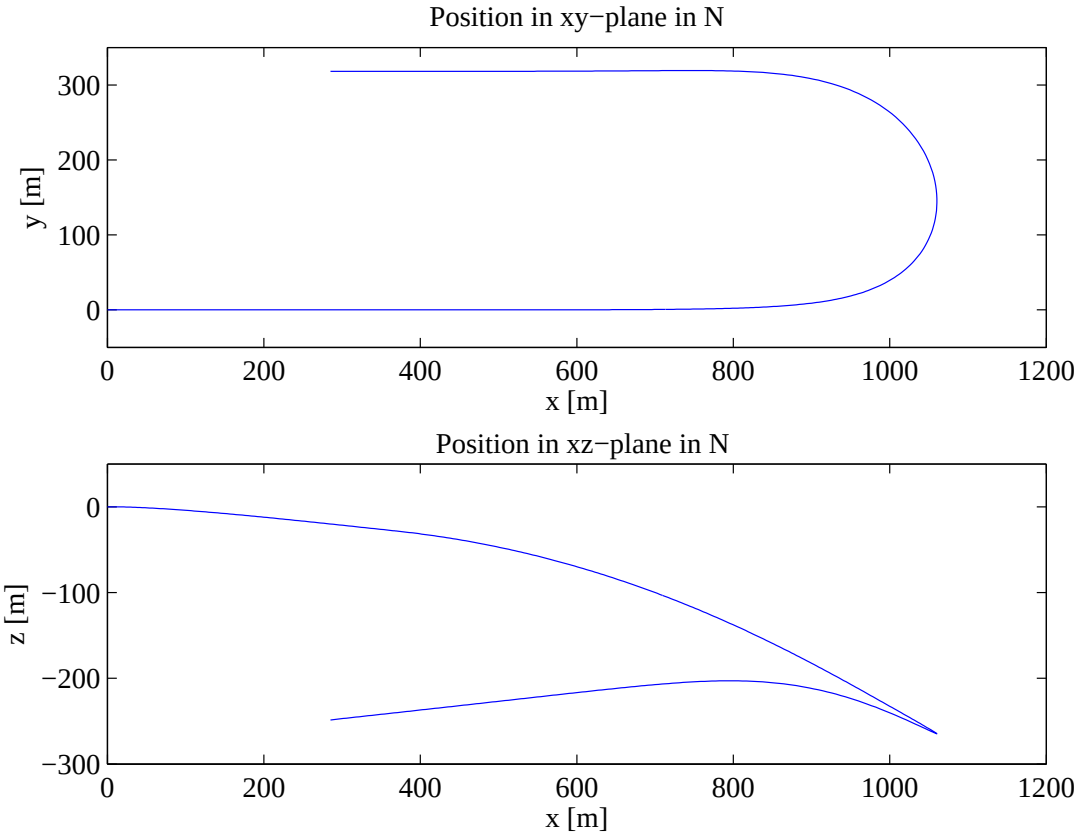


Figure 7.8: *The top graph show the position of the aircraft in the ${}^{\mathbb{N}}\boldsymbol{x}^{\mathbb{N}}\boldsymbol{y}$-plane, and the bottom graph show the aircraft in the ${}^{\mathbb{N}}\boldsymbol{x}^{\mathbb{N}}\boldsymbol{z}$-plane.*

Figure 7.9 show the deflection angles on the aircraft during a 180° rotation in yaw. It can be seen that no of the actuators is saturated.

7.4 Controller Performance

The aircraft model is linearized using a nonlinear feedback linearization, and this type of linearization require a good model, because nonlinearities are canceled out. The system would then appear as a linear system after the cancellation. The controller is therefore sensitive toward

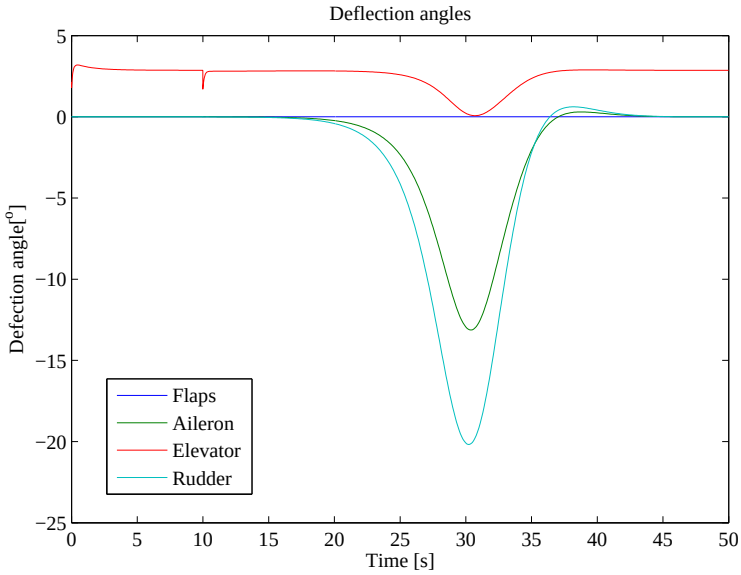


Figure 7.9: *Deflection angles of the aircraft in a 180° rotation in yaw.*

modeling errors. An example on a model error would the moment of inertia $\boldsymbol{I}$. The aircraft is stabilized using two feedback loops, one controlling the angular velocity, and one controlling the orientation of the aircraft. This make a more robust control system.

A simulation is made, where the aircraft turn 180° in yaw. The $\boldsymbol{I}$ is scaled with 0.2 in order to introduce a model error. Measurement noise is also added in the form of noise on the gyros.

${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ is measured with the GyroCube. A test flight has been performed and this test shows the measurement noise on the gyros, see chapter 9 on page 72. This test shows the variance of the noise on ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, when the aircraft is in the air, to be less than 0.02. To perform the test is noise added with a variance on $0.4\frac{rad}{s}$ added to ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, see figure 7.10.

Figure 7.11 shows ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$, and it can be seen that the controller still reach the correct orientation, but it is seen that there is a steady state error due to the noise in the system. The simulation shows that the controller can handled the noise from the gyro, and model errors like a wrong moment of inertia. The noise from the gyros would be reduced on the real aircraft, because a kalman filter would be implemented. Chapter 8 contains a stability analysis of the aircraft, which checks stability of the aircraft.

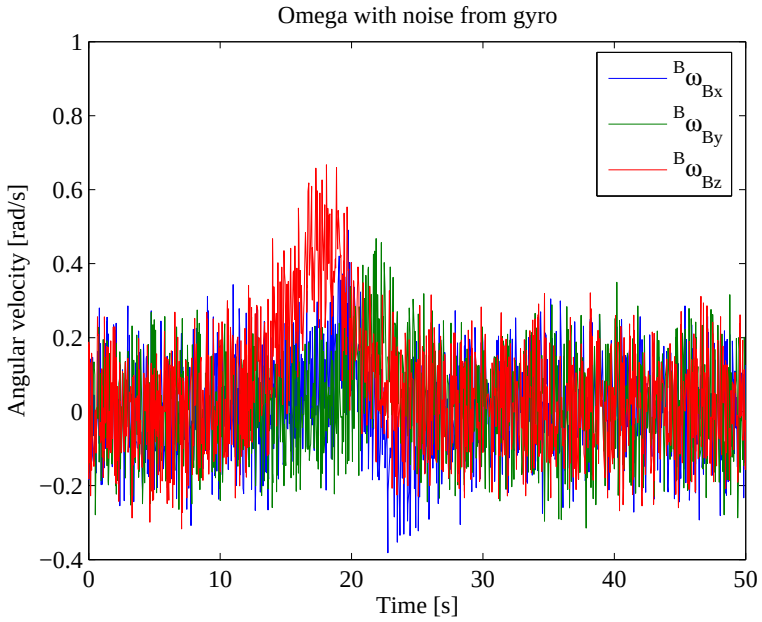


Figure 7.10: *$^{\mathbb{B}}\boldsymbol{\omega}$ where noise is added with an 0.4$\frac{rad}{s}$ variance.*

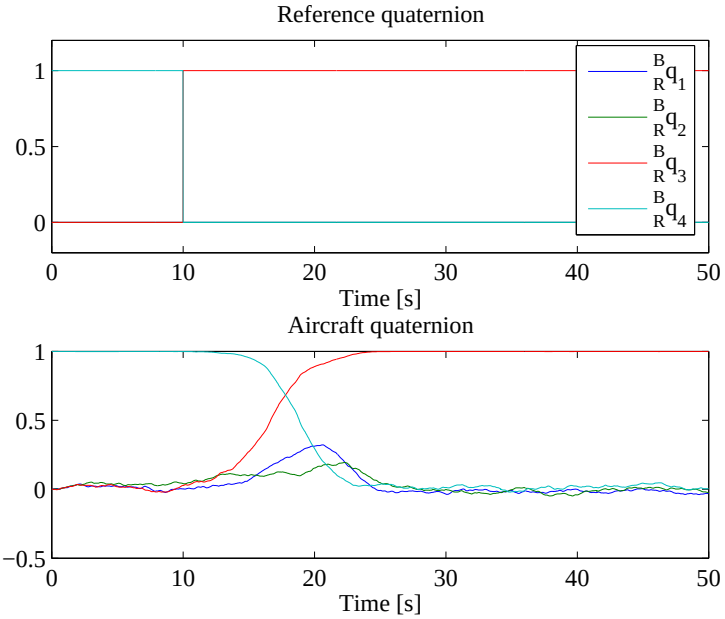


Figure 7.11: *The figure show $^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$, where noise is added to $^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$.*

Chapter 8

Stability

This chapter contains a stability analysis of the system. We use Lyapunov's stability criteria [7], to prove that the aircraft including controller is stable. Since the controller is designed such that the system is input-output stable, this chapter only contains the stability analysis of the internal dynamics.
The analysis is conducted using Matlab symbolic toolbox, which eases the task and reduces the likely hood of human errors. The analysis in this chapter is presented through the theory with results from the stability.m Matlab script on the Project CD-rom. This is done to minimize the complexity of the formulaes in this chapter and to give a better understanding of why the aircraft is stable.

A system is in general stable in a equilibrium point $\boldsymbol{x}_0$, if all solutions to the system starting nearby an equilibrium stays nearby the equilibrium. It is asymptotically stable, if the solutions tends to the equilibrium as time goes to infinity, otherwise it is unstable [7, p. 112].

We look at Lyapunov's Direct Method, to check whether the aircraft is stable or not, which states that a system (8.1),

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) \tag{8.1}$$

is stable or asymptotically stable, if there exists any positive definite function $V(\boldsymbol{x})$ and $V(\boldsymbol{0}) = 0$, such that $\dot{V}(\boldsymbol{x})$ is negative semi-definite or negative definite.

Unfortunately the aircraft model is rather large and complex in its structure, and therefore is the Direct Method difficult to apply. Instead we will use Lyapunov's Second Method or the Interconnected Systems approach, which divides the problem into smaller systems to reduce the complexity.

8.1 Interconnected Systems Approach

Since the controller are designed such that the system is input-output stable the only part left is to check the stability of the internal dynamics $\dot{\boldsymbol{\eta}}$[1]. This is quite important, since we have to know

[1] Internal dynamics is the dynamics with no measured output and typical not controlled.

what happens to the remaining dynamics in the aircraft, when we control the angular rates. In worst case, if they were unstable and the aircraft could crash or become uncontrollable.

We start with the system on the normal form(7.20). The normal form in it standard form presents the internal dynamics on the form (8.2),

$$\dot{\boldsymbol{\eta}} = f(\boldsymbol{\eta}, \boldsymbol{\zeta}) \tag{8.2}$$

$$\boldsymbol{\eta} = \begin{bmatrix} {}^{\mathbb{B}}v_{\mathbb{B},x} \\ {}^{\mathbb{B}}v_{\mathbb{B},y} \\ {}^{\mathbb{B}}v_{\mathbb{B},z} \\ [{}^{\mathbb{B}}_{\mathbb{I}}q]R1 \\ [{}^{\mathbb{B}}_{\mathbb{I}}q]R2 \\ [{}^{\mathbb{B}}_{\mathbb{I}}q]R3 \\ [{}^{\mathbb{B}}_{\mathbb{I}}q]R4 \end{bmatrix} \tag{8.3}$$

$$\boldsymbol{\zeta} = \begin{bmatrix} {}^{\mathbb{B}}\omega_{\mathbb{B},x} \\ {}^{\mathbb{B}}\omega_{\mathbb{B},y} \\ {}^{\mathbb{B}}\omega_{\mathbb{B},z} \end{bmatrix} \tag{8.4}$$

where $\boldsymbol{\zeta}$ is the input controlled feedback linearised states. And $\boldsymbol{\eta}$ are the remaining not directly measured and non controlled dynamics.

The aircraft model is different from a system on the standard normal form, since it has multiple inputs, and each input controls more than one state. Thus the normal form used in this project is slightly different (8.5) than the standard.

$$\dot{\boldsymbol{\eta}} = f(\boldsymbol{\eta}, \boldsymbol{\zeta}) + g(\boldsymbol{\eta}, \boldsymbol{\zeta})\boldsymbol{u} \tag{8.5}$$

$g(\boldsymbol{\eta}, \boldsymbol{\zeta})$ in (8.5) is the input influence on the internal dynamics. Adding this term actually causes some stability problems, because when we feedback linearise the system we introduce a input signal (7.7), which cancels the nonlinear terms in the controlled state equations. In fact this adds the same nonlinear terms in the remaining states, which could cause the internal dynamics to be unstable.

The introduced feedback linearising control input is $\boldsymbol{u} = \alpha(\boldsymbol{\eta}, \boldsymbol{\zeta}) + \beta(\boldsymbol{\eta}, \boldsymbol{\zeta})\boldsymbol{v}$, where $\boldsymbol{v}$ is the new input signal. To make sure that the internal dynamics are stable we introduce the feedback law with a zero input $\boldsymbol{v} = 0$, thus revealing (8.6).

$$\dot{\boldsymbol{\eta}} = f(\boldsymbol{\eta}, \boldsymbol{\zeta}) + g(\boldsymbol{\eta}, \boldsymbol{\zeta})\alpha(\boldsymbol{\eta}, \boldsymbol{\zeta}) \tag{8.6}$$

Due to the complexity of (8.6), we look at the system as an interconnected system. Thus we divide the system into subsystems with non-interconnected terms $f_i(\eta_i)$ and interconnected terms $g_i(\boldsymbol{\eta}, \boldsymbol{\zeta})$, where $i = 1 \ldots m$, and m is the number of the zero-dynamic states.

$$\begin{aligned} \dot{\eta}_i &= f_i(\eta_i) + g_i(\boldsymbol{\eta}, \boldsymbol{\zeta}) \\ f_i(\eta_i) &= f(\eta_i, \mathbf{0}) + g(\eta_i, \mathbf{0})\alpha(\eta_i, \mathbf{0}) \\ g_i(\boldsymbol{\eta}, \boldsymbol{\zeta}) &= f(\boldsymbol{\eta}, \boldsymbol{\zeta}) + g(\boldsymbol{\eta}, \boldsymbol{\zeta})\alpha(\boldsymbol{\eta}, \boldsymbol{\zeta}) - f_i(\eta_i) \end{aligned} \tag{8.7}$$

We check that the system is stable by looking at Lyapunov's Second Method, which states that if there exist $i = 1..m$ positive definite functions $V_i(\eta_i)$ and $V_i(0) = 0$, such that

$$\dot{V}_i(\eta_i) = \frac{\partial V(\eta_i)}{\partial \eta_i} \cdot f_i(\eta_i) \leq -\alpha_i \phi_i^2(\eta_i) \tag{8.8}$$

$$\left\| \frac{\partial V(\eta_i)}{\partial \eta_i} \right\| \leq \beta_i \phi_i(\eta_i) \tag{8.9}$$

where α_i, and β_i in (8.8) and (8.9) are positive constants and $\phi_i(\eta_i)$ are a continuous and positive definite functions.

And if the interconnection terms are bounded such that they satisfy (8.10),

$$\|g_i(\boldsymbol{\eta}, \boldsymbol{\zeta})\| \leq \sum_{j=1}^{m} \gamma_{i,j} \phi_j^2(\eta_j) \tag{8.10}$$

where $\gamma_{i,j}$ in (8.10) is a non-negative constant. Then Lyapunov's theory conclude that the overall system is asymptotically stable[7].

8.2 The Stability Analysis

To perform the stability analysis using the Lyapunov method presented in section 8.1 we will start by finding the different Lyapunov candidate functions $V_i(\eta_i)$. A good rule of thumb is to use an energy function or the squared state as a candidate function. Thus we try following candidates:

$$V_i(\eta_i) = \frac{1}{2}\eta_i^2 \tag{8.11}$$

which is a positive definite function, and $V_i(0) = 0$.

The first step following the Interconnected system approach is to check that the non-interconnected terms are stable, by calculating the derivative of (8.11) over the solution trajectory of the system. Thus applying (8.8) on the Lyapunov candidates (8.11).

$$\dot{V}_1(\eta_1) = \eta_1(c_0 - c_1\eta_1^2 - (c_2 + c_3\eta_1)\eta_1 + (c_4 + c_5\omega_p)\omega_p) \tag{8.12}$$

$$= c_0\eta_1 - c_1\eta_1^3 - c_2\eta_1^2 - c_3\eta_1^3 + c_4\omega_p\eta_1 + c_5\omega_p^2\eta_1 \tag{8.13}$$

where the constants $c_{0..5}$ are positive constants, and the angular velocity of the propel ω_p is positive. Then the first term (8.12) shows that the first state is stable. Since $\eta_1 = {}^{\mathbb{B}}v_x$ is positive[2], it is seen that the aircraft over time tends towards a positive velocity proportional to ω_p, thus it tends toward an equilibrium and is therefore asymptotically stable.

We skip the next state $\eta_2 = {}^{\mathbb{B}}v_{\mathbb{B},y}$, which is described in section 8.2.2 instead, since this eases the understanding of the stability proof for the remaining states.

To show that $\eta_3 = {}^{\mathbb{B}}v_z$ is stable we use the same approach as on (8.12), thus reveiling (8.14)

$$\dot{V}_3(\eta_3) = \eta_3(9.82 - 0.0195\eta_3) \tag{8.14}$$

[2]We expect the aircraft to move forward all the time.

which is stable since the state tends toward a constant level. This is an interesting discovery that at some point in time if the aircraft falls downward it will reach a maximum speed, at least from this part of the dynamic equations. However this stability isn't that satisfying, because what Lyapunov's methods don't tell is that there are more important stability factors in this state.

Think of the aircraft as a thin piece of paper. If it is dropped it will fall, but with different velocity depending on the drag surface area of the paper. If it is flat[3] in the air it falls slowly otherwise it falls faster. Now instead of having a piece of paper with an even mass distribution we make it nose heavy(like an aircraft). The result is that the paper, or aircraft, will turn in the air inducing a forward velocity. It still falls down but the velocity in the z-axis decreases and increases in the x-axis.

This leads us to the stability of a high winged aircraft[4], namely that the increase in forward velocity ${}^{\mathbb{B}}v_x$ generates lift, and pitch moment, thus turning the aircraft again, and at some point forcing the aircraft to stop falling, thus it is capable of gliding in the air, without engine thrust. It will eventually reduce its forward velocity and start falling again, but at a slower rate. The motion of the stable high winged aircraft is shown on figure 8.1, where the situation is simulated using the implemented aircraft model. It should be noted that the simulation result on figure 8.1,

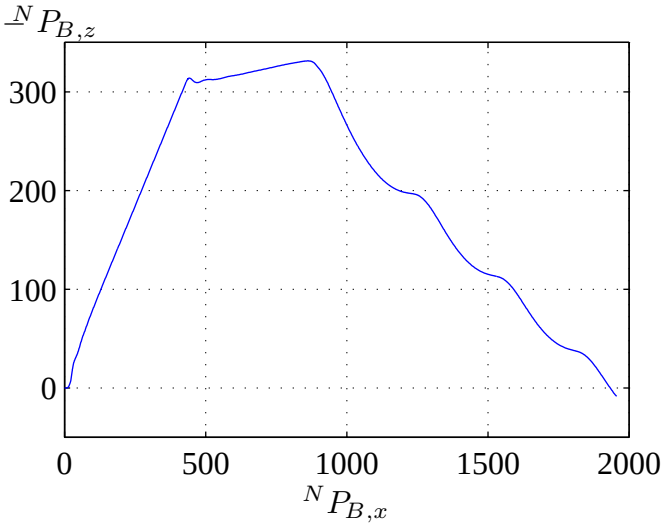


Figure 8.1: *The simulated position of the aircraft during a fall, showing the stability, where the aircraft glides in the air, which causes it to slowdown.*

is created by using a controller to reach the altitude around ${}^{\mathbb{N}}P_{\mathbb{B},z} = -320$m. Where the aircraft is stabilised. At ${}^{\mathbb{N}}P_{\mathbb{B},x} = 900$m the control output is frozen and the engine is shut off $\omega_p = 0$. After the engine shut off the aircraft starts falling, and goes in to this stable cycle where it at some point reaches an equilibria where it glide with a steady velocity in both forward and downward directions.

The remaining states are quite different from the two above. Because they only get the stability introduced through the interconnection between other states. Thus they cannot be asymptoti-

[3]The surface is tangential with the floor.

[4]A high winged aircraft is an aircraft where the main wing is located above the CoG.

cally stable except if the interconnected state are asymptotically stable, because they inherits stability from other states. One demand for a state to do so, is that the interconnection terms are bounded (8.10), which they are if the states are bounded. Thus the remaining states are stable, since ζ is controlled such that it is bounded and asymptotically stable. And the two states described above are asymptotically stable and bounded.

8.2.1 The Quaternion

The quaternion is stable, because it inherits stability from the controlled angular velocity of the aircraft.

$$\frac{\mathrm{d}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}}{\mathrm{d}t} = \frac{1}{2}\begin{bmatrix} \boldsymbol{S}({}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}) & {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \\ {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}^{\mathrm{T}} & 0 \end{bmatrix} \tag{8.15}$$

looking at (8.15), where S is the skew symmetric matrix defined in (6.12), shows that if there is a angular velocity ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}} \neq \mathbf{0}$ then the quaternion changes. If there are no angular velocity, hence the controlled states are in the equilibrium ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B},0} = \mathbf{0}$, then the quaternion is steady and stable. Further more it is noted that by definition the quaternion is bounded $\|\boldsymbol{q}\|_{=}1$ see appendix C, thus helping to ensure stability.

8.2.2 Y-Velocity Stability

As for the quaternion the state $\eta_2 = {}^{\mathbb{B}}v_y$ is stable due to the influeces from the interconnected terms. By looking at the force which causes the sidewise velocity of the aircraft (8.16). We find that it is dependent on: the sideslip angle β, the roll rate ${}^{\mathbb{B}}\omega_x$, and yaw rate ${}^{\mathbb{B}}\omega_y$ of the aircraft see section 5.1.5 on page 28.

$${}^{\mathbb{A}}F_{y,aero} = \bar{q}S\left(\boldsymbol{P}^{\mathrm{T}}_{{}^{\mathbb{A}}F_{y,zero}}\begin{bmatrix}\beta_{\mathbb{A}} & \frac{b}{2V_\infty}{}^{\mathbb{A}}\boldsymbol{\omega}_{\mathbb{B}}\end{bmatrix}^{\mathrm{T}} + \boldsymbol{P}^{\mathrm{T}}_{{}^{\mathbb{A}}F_{y,input}}\boldsymbol{\delta}\right) \tag{8.16}$$

It is also noticed that $\boldsymbol{P}_{{}^{\mathbb{A}}F_{y,zero}}$ is negative with regard to the influence from β, which is a dampening of the force. Thus the state is stable, if $\boldsymbol{\omega}_{\mathbb{B}}$ is at its equilibria, which our inner control loop forces it to be.

The stability in the Y-velocity is simulated as the example on figure 8.1, which results in the trajectory of the aircraft shown on figure 8.2. It should be noted that the simulation result on figure 8.2, is created by using a controller to reach the altitude around 300m, thus ${}^{\mathbb{N}}P_{\mathbb{B},t=0} = \begin{bmatrix}0 & 0 & -300\end{bmatrix}$. After this state is reached, is the aircraft controlled, such that it has 25° roll angle and 45° pitch angle. At ${}^{\mathbb{N}}P_{\mathbb{B},x} = 300$m is the controller output frozen and the engine is shut off $\omega_p = 0$. The outcome is that the aircraft accelerates downwards, and starts to build up lift as on figure 8.1. And goes into a stable cyclic motion toward the earth, where ${}^{\mathbb{B}}v_{\mathbb{B},z}$ slowly decreases to a constant value. It is also seen that the radius of the cyclic path is constant telling us that the sidewise velocity does not change.

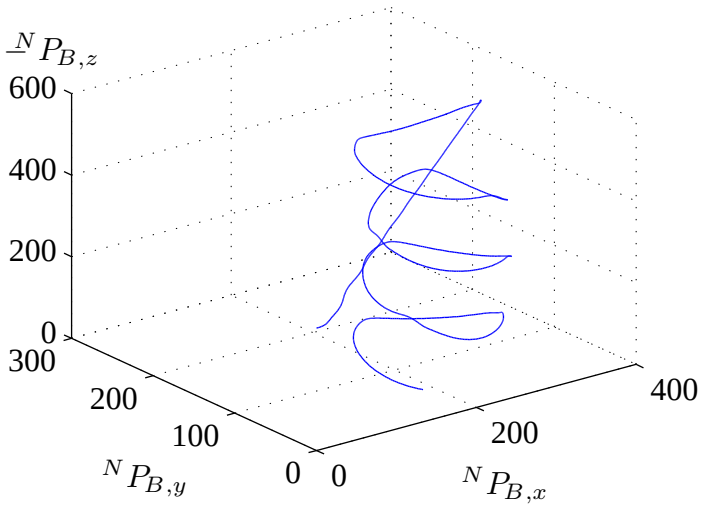


Figure 8.2: *The simulated position of the aircraft during a fall, showing the stability where the aircraft glides in the air, which causes it to slowdown.*

8.3 The Result

From the conducted stability analysis we conclude, that the aircraft is assymptotical stable, as long as α and β is non-singular, which they are within the working range of the model.

Is is also shown that for the aircraft to be stable, is it required that the angular rates ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ are controlled, such that they are bounded and assymptotical stable[5].

[5]The assymptotical requierement is not necessary for stability, however it is prefered from a control point of view.

Chapter 9

Test Flight

Three test flights has been performed, and the purpose with the tests is to test the aircraft, hardware, software, and validate the aircraft models. The three test flights is described and the results is shown in this chapter.

The models in chapter 5 and 6 need to be validated, and this is done by test flights. During the tests are data from the GPS, the GyroCube, and the servo board logged using the data logger, see appendix A on page 81. During a test is it possible to measure the following data:

- The GPS provides:
  - The position ${}^{\mathbb{E}}\boldsymbol{P}_{\mathbb{B}}$.
  - The velocity ${}^{\mathbb{E}}\boldsymbol{V}_{\mathbb{B}}$.
- The gyroCube provides:
  - The angular velocity ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$.
  - The translatory acceleration ${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}}$.
- The servo board can measure the position of the following servos:
  - Aileron δ_a.
  - Elevator δ_e.
  - Rudder δ_r.
  - Flaps δ_f.
  - Test indicator is an extra receiver channel used to indicate, when a test is conducted. This helps to identify each experiment.
  - Manuel/autonomous mode is also a signal from a switch, which the pilot control.

The test flights have three purposes, get the aircraft approved, test that hardware and software work properly when the aircraft is flying, and validate that the aircraft models are correct.

9.1 Test Flights

The test flights toke place the 23. of May 2005 on Pandrup-mfk, where the aircraft was in the air for the first time. The aircraft was in the air three times.

9.1.1 First Test Flight

The aircraft have a total weight above 7kg, and this mean that it is necessary to get the aircraft approved. This mean that a certified person has to check the aircraft and check that the pilot can fly the aircraft. The first test flight was done in order to get this approval. Only the vital electronic was turned on, which include, the servomotors, servo board, and receiver. The PC-104 and the sensors was turned off. The first test flight went well and the aircraft got the approval.

9.1.2 Second Test Flight

The purpose with the second flight is to test sensors, and servo board. This mean that the GPS, the gyroCube, and the servo board can sample a meaningful value with low noise and send it to the PC-104, where it is saved in a file, while the aircraft is in the air, and everything is turned on. The GyroCube and the servo board worked fine and the result of the test can be seen on figure 9.1 and figure 9.2. The GPS module didn't measured any positions or velocity. The reason for this could be the GPS antenna, isn't the original antenna for the gps receiver, and it can therefore have a problem with reaching any satellites at all. Figure 9.1 shows the data

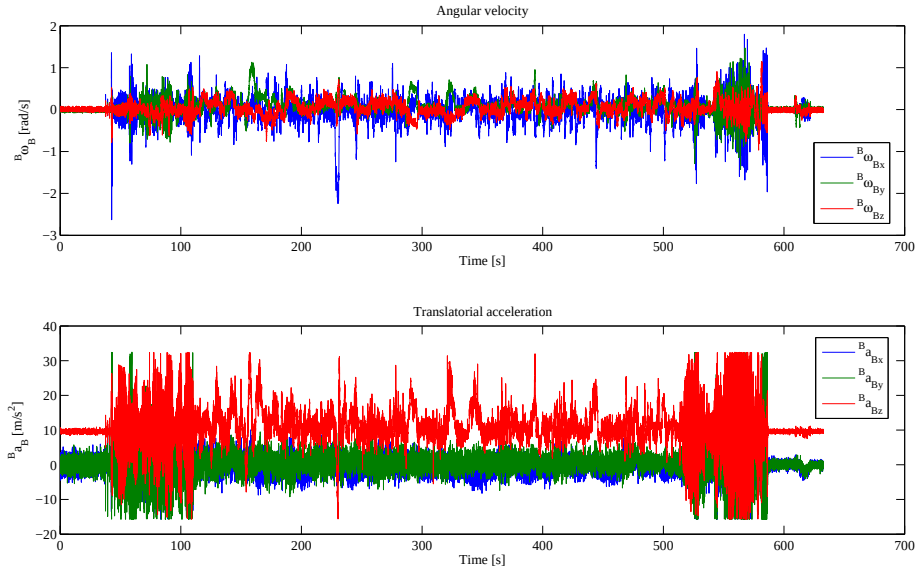


Figure 9.1: *The graph show the result from the GyroCube on the second test flight. The graph show ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ on the top graph and ${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}}$ on the bottom graph.*

logged from the GyroCube, which are ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ and ${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}}$. The test lasted about 650s, and there is very little noise on the signal in the start and in the end where the aircraft stand still, but especially the accelerometers go in saturation during takeoff (40s-110s) and landing (510s-590s). If these sensors should be used under these circumstances should they be shifted with a sensor with a larger working area, or a filter should be implemented. It is concluded that the output from the GyroCube can be used, because it is mainly used when the aircraft is in the air (110S-510s), not under takeoff or landing.

Figure 9.2 show the deflection angles on flaps, aileron, elevator, and rudder. Another channel is used as indicator signal, which isn't used in this test. It can be seen on the signals, that it is possible to read the deflection angles on aileron, elevator, and rudder. The flaps are connected to a contact on the transmitter, and it can only take two values as it is shown on the graph.

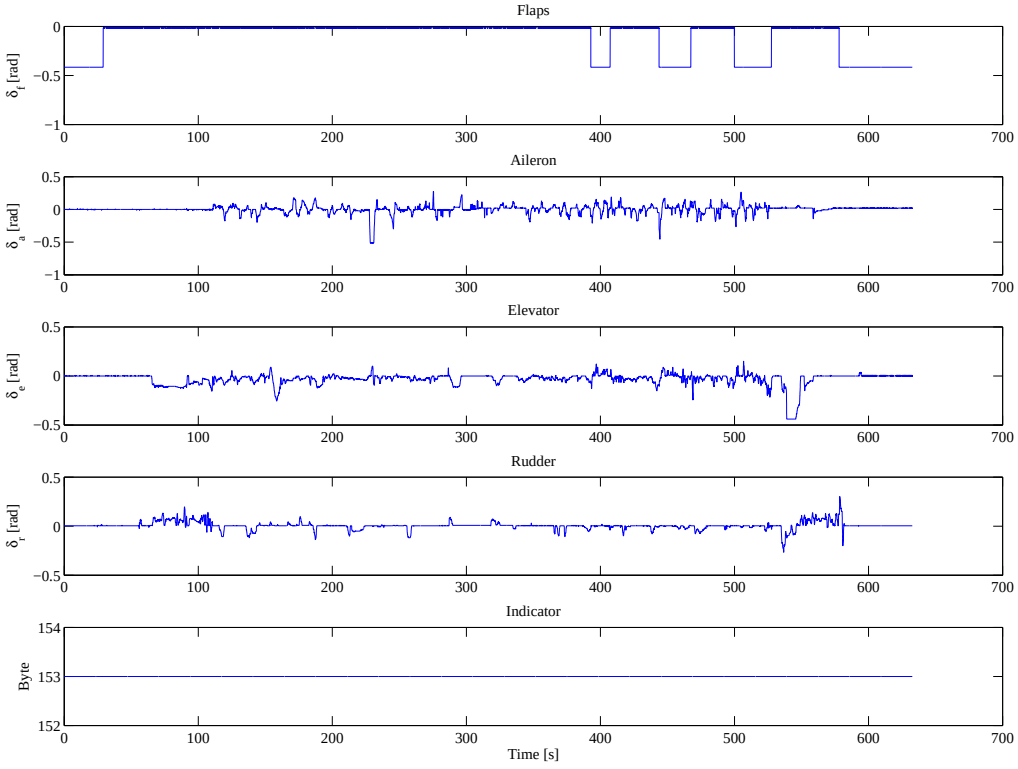


Figure 9.2: *The graph show the deflection δ_f, δ_a, δ_e, and δ_r, for the second test flight. Also the Indicator signal is shown.*

9.1.3 Third Test Flight

The purpose with the third test flight is to validate the aircraft models. The test is performed by making steps on aileron, elevator and rudder. Each time a test is made, is the indicator signal

high. Due to a loose window, where the pilot was forced to land early, thus it was only possible to conduct the three following tests:

1. Elevator is used during the takeoff.
2. A fast roll is made to the right using aileron.
3. A fast roll is made to the left using aileron.

This section looks at the two roll maneuverer. Figure 9.3 show the servo signals given to the system in the period from 178s to 190s after the test start. The signal on the flaps is noise. It is seen on the aileron that a step has been applied to the system first in one direction. The aircraft is then stabilized and a step is given in the other direction. Elevator and rudder is used to stabilize the aircraft, between the maneuvers.

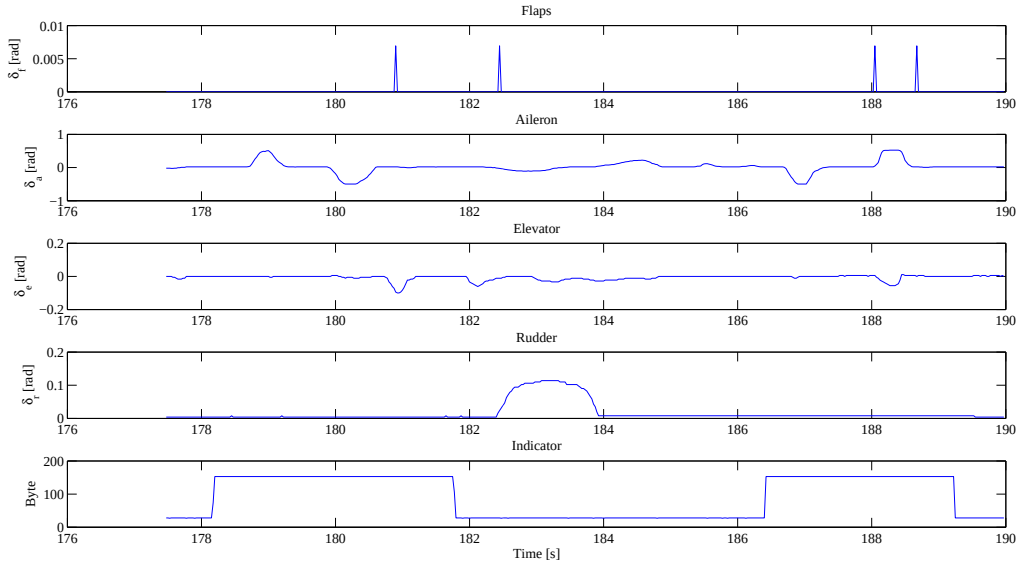


Figure 9.3: *The figure shows the deflection angles of aileron, elevator, and rudder. The indicator signal show when the roll maneuverer are made.*

Figure 9.4 shows ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$. The top graph show the measured angular velocity, while the bottom graph show a simulation. The input for this simulation is the deflection angles δ_a, δ_e and δ_r, so the aircraft simulator get the same input as the aircraft. Figure 9.4 shows that ${}^{\mathbb{B}}\omega_{\mathbb{B},x}$ react the same way for both the real model, and the simulator. The difference is that the simulator don't react so fast as the real model and the amplitude on the signal is not so high in the start of the simulation. ${}^{\mathbb{B}}\omega_{\mathbb{B},y}$, and ${}^{\mathbb{B}}\omega_{\mathbb{B},z}$ act more different, where especially ${}^{\mathbb{B}}\omega_{\mathbb{B},y}$ drift away from the signal. There can be several reason to that the two components don't match, in simulation and test flight. $\boldsymbol{V}_\infty$ was not measured in the test flight because the GPS did not get any position and velocity. This difference of $\boldsymbol{V}_\infty$ in simulation and test flight would generate different forces, and torques, which courses the aircraft and simulator to act different. Another error can be

the moment of inertia. The results indicate the inertia for the model is to high because the simulation don't react as fast as the real aircraft.

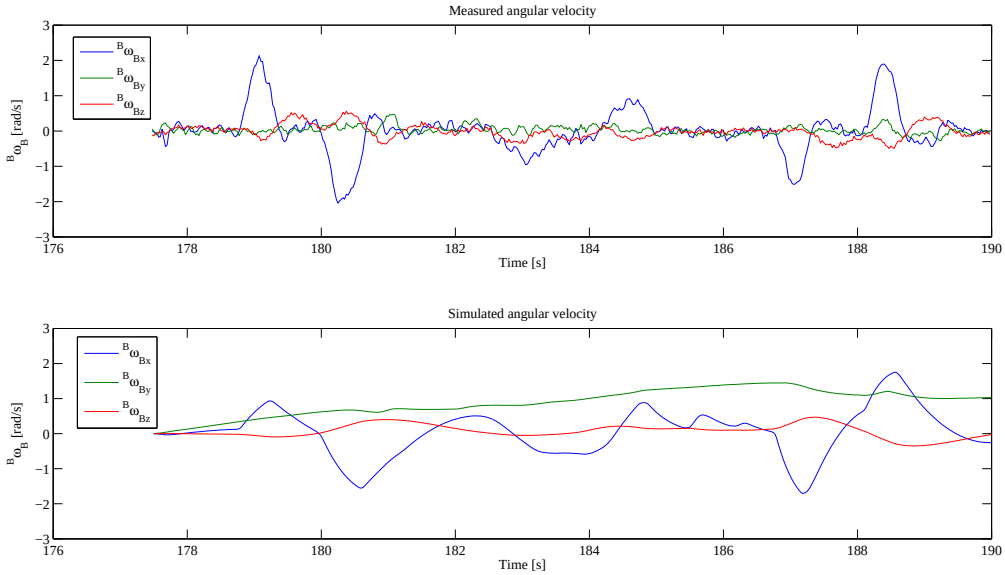


Figure 9.4: *The figure show the measured and simulated values of* ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$.

Overall, this test shows that the simulator react as the aircraft in ${}^{\mathbb{B}}\omega_{\mathbb{B},x}$ but it is necessary with more test to find the moment of inertia, and validate the aircraft models for pitch and yaw. It is necessary with a test flight where steps on elevator and rudder are performed.

Chapter 10

Conclusion

The purpose of the project is to model and control an aircraft, and the goal is to get the aircraft to fly autonomously and to control the orientation of the aircraft. The subjects covered in this project are: modeling the aircraft, controlling the aircraft, design of a hardware platform for the Cessna Skylane 182 aircraft model, and implementation of a data logger.

Four models has been derived: an aerodynamic model, a propeller model, a dynamic model, and a kinematic model. All models are implemented in a flight simulator. This flight simulator shows a 3D presentation of the aircraft with respect to the world ($\mathbb{N}$). It is then possible to visualize how the aircraft reacts to different inputs. It has therefore been possible for a pilot, which is experienced in flying model aircrafts, to validate the model. The conclusion of this test was that the aircraft model reacts as it is supposed to do. The result from the conducted test flights shows that the model reacts as the real aircraft in roll. It shows also that more tests are required to validate the model, and to make a system identification of the uncertain parameters in the model.

A nonlinear feedback controller for MIMO systems was designed to stabilize ${}^{\mathbb{B}}\boldsymbol{\omega}_B$. An outer control loop is controlling the orientation of the aircraft using quaternions. These two controllers are implemented on the flight simulator. It is then possible both to stabilize ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, and control ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$. It is possible to give the controller a reference, and it will follow the most time efficient path, in order to reach and track the wanted orientation. The controllers are cascade coupled, which makes the aircraft more robust towards noise and model errors. Which has been tested, and the result shows that the controller structure is robust toward measurement noise and parameter changes. Neither the velocity controller or the orientation controller was implemented on the actual aircraft.

The feedback controller contains internal dynamic, which is uncontrolled. A Lyapunov stability analysis was conducted successfully to check whether the internal dynamics were stable, and the conclusion of the analysis is that the internal dynamics are asymptotically stable.

The aircraft has been instrumented with sensors, actuators, interfaces, batteries, camera, and flight computer. All modules on this platform works as they are supposed to, except the GPS antenna. The GPS module work as it is suppose to do, but the antenna don't match with the module. A data logger has been implemented on the platform, and test flights shows, that it is

possible to log deflection angles for: flaps, aileron, elevator, and rudder. It was also possible to log ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ and ${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}}$ from the GyroCube. The data logger can also log positions and velocities from the GPS receiver with the proper GPS antenna. It can be concluded that the data logger and hardware platform works as they are supposed to.

It is always possible for the pilot to control the aircraft, and always gain the control from the flight computer if needed. Data can be logged from the sensors, and saved on the harddisk, and the data will not be lost, even if the flight computer stops. The effect from vibrations in the aircraft are also minimized in the construction.

The overall conclusion of the project is that a working test platform is made, where it is possible to log usable test data. The roll part of the aircraft model is validated, during a test flight. It is possible to stabilize ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ and control the orientation of the aircraft in a 3D simulator, and it is proved that the controller and the aircraft are asymptotically stable.

10.1 Perspective

This section lists the improvements, which has to be done in order to make the aircraft fully autonomously.

The following suggest improvements which are necessary for the aircraft to keep a specific orientation, or improvements which make it easier to operate the aircraft.

- The aircraft model needs to be validated, and this includes a series of test flights.The moment of inertia should be found through experiments, because it is observed that the current moment of inertia matrix is to big.
- A Kalman filter should be implemented to filter the input signals from GPS and GyroCube, and to implement sensor fusion between the redundant sensors.
- With the current controller is the aircraft able to follow an orientation reference. If should follow a path, or keep a curtain position, must a translatory kinematic model be derived, together with a path generator.
- A new GPS antenna, which match the GPS board should be acquired, so it is possible to get in contact with satellites.
- Use a Ethernet card supported by XPC instead of RS232 connection to communicate with the flight computer to get higher download speed.

If the aircraft should be truly autonomous should the following be implemented.

- The aerodynamic model describes how the aircraft reacts when it is flying in the air. If the aircraft should be able to takeoff and land autonomous, must ground effect be included in the aerodynamic model.

- A path generator can be implemented, which make it possible for the aircraft to follow a path.
- If the aircraft should be able to land autonomously, must it be possible to control flaps, and throttle for the main engine. In this case is a redesign of the servo board needed.
- ω_P should be measured, such that it is possible to control the generation of thrust.
- The accelerometers should be change with a type with a larger working area, if the aircraft should be able to perform takeoff and landing, using this sensor.
- A secondary and more accurate altitude sensor is needed during takeoff and landing, laser or ultrasonic would be preferred in this case.

Bibliography

[1] *MathWorld.com*. Wolfram Research, Inc, 2004. URL:http://www.mathworld.com/.

[2] A. T. Andersen, P. Bak, M. A. Hansen, R. Jensen, M. H. Sørensen, and J. Xie. Autonomous model airplane. Technical report, Department of control Engineering, AAU, 2004.

[3] N. G. R. Center. *Beginner's Guide to Aerodynamics*. 2004. www.grc.nasa.gov/WWW/K-12/airplane.

[4] E. European Space Agency. Navigation, 2004. http://www.esa.int/EGNOS.

[5] M. Hepperle. Aerodynamics for model aircraft, 2004. http://mh-aerotools.de/airfoils.

[6] A. Isidori. *Nonlinear control systems*. Springer-verlag, 3 edition, 1995.

[7] H. K. Khalil. *Nonlinear systems*. Prentice Hall, 3rd edition, 2000.

[8] D. W. H. mason. *Applied Computational Aerodynamics*.

[9] Mathworks. Matlab guide, 2004. http://www.mathworks.com.

[10] D. Megginson. Megginson technologies, 2005. http://www.megginson.com/.

[11] Novatel. L1 gps firmware, 2004. www.novatel.com/Documents/Manuals/om-20000086.pdf.

[12] Novatel. Superstar ii, user manual, 2004. http://www.novatel.ca/Documents/Manuals/om-20000077.pdf.

[13] B. N. Pamadi. *Performance, Stability, Dynamics, and Control of Airplanes*. American Institute of Aeronautics and Astronautics, Inc., 1998. ISBN 1-56347-222-8.

[14] J. Roskam. *Airplane Flight Dynamics and Automatic Flight Controls*. 1995.

[15] J. R. Wertz. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, 1978.

Appendix A

Data Logger

This appendix describe how the data logger is designed. The data logger has three parts, a gyro logger, a GPS logger, and a servo logger. The software in the PIC processor on the servo board is also explained here. The last section in the appendix is an user manual to the data logger.

The aircraft models need to be validated in test flights. These test flights can also be used to test controller types. In order to validate the models and test controllers, it is necessary to log the output from the various sensors on the aircraft. This chapter describes how the data logger is designed. The data logger is placed on the PC-104, and log data from the gyros, the GPS, and the servomotors.

It is a requirement that the data is logged in real time, so the operation system must full fill this requirement. It is chosen to use the Matlab toolbox XPC to implement the data logger on the PC-104. A small real time operation system is installed on the PC-104, also called the target. It is then possible to make a program on another computer with Matlab and Simulink installed. The program is implemented in Simulink and then sent to the target, where it is executed in real time. Drivers for the sensors are implemented in Simulink S-functions, so it is possible to log the outputs from the sensors.

The data logger consist of three main parts which are: a GPS logger, a gyro logger, and a servo logger. Figure A.1 show what hardware the different parts of the data logger is connected to. This chapter focus on the design of the gray boxes in figure A.1.

Gyro logger log the angular velocity ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$ of the aircraft, and the translatory acceleration ${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}}$ of the aircraft. The velocity and acceleration is measured with the GyroCube. The output from the GyroCube is an analog signal, which is sampled with the A/D-converter. The output from the samplings board is a 10 bit signal.

GPS logger logs the position of the aircraft in ECEF coordinates ${}^{\mathbb{E}}\boldsymbol{P}_{\mathbb{B}}$, and the velocity ${}^{\mathbb{E}}\boldsymbol{v}_{\mathbb{B}}$ of the aircraft with respect to $\mathbb{E}$. The GPS logger gets data from the GPS via a RS232 connection, and it must send commands to the GPS receiver.

Servo logger logs an eight bit parallel signal from the servo board via the samplings board, when the aircraft is in manual mode. This byte represent the position of the servomotors.

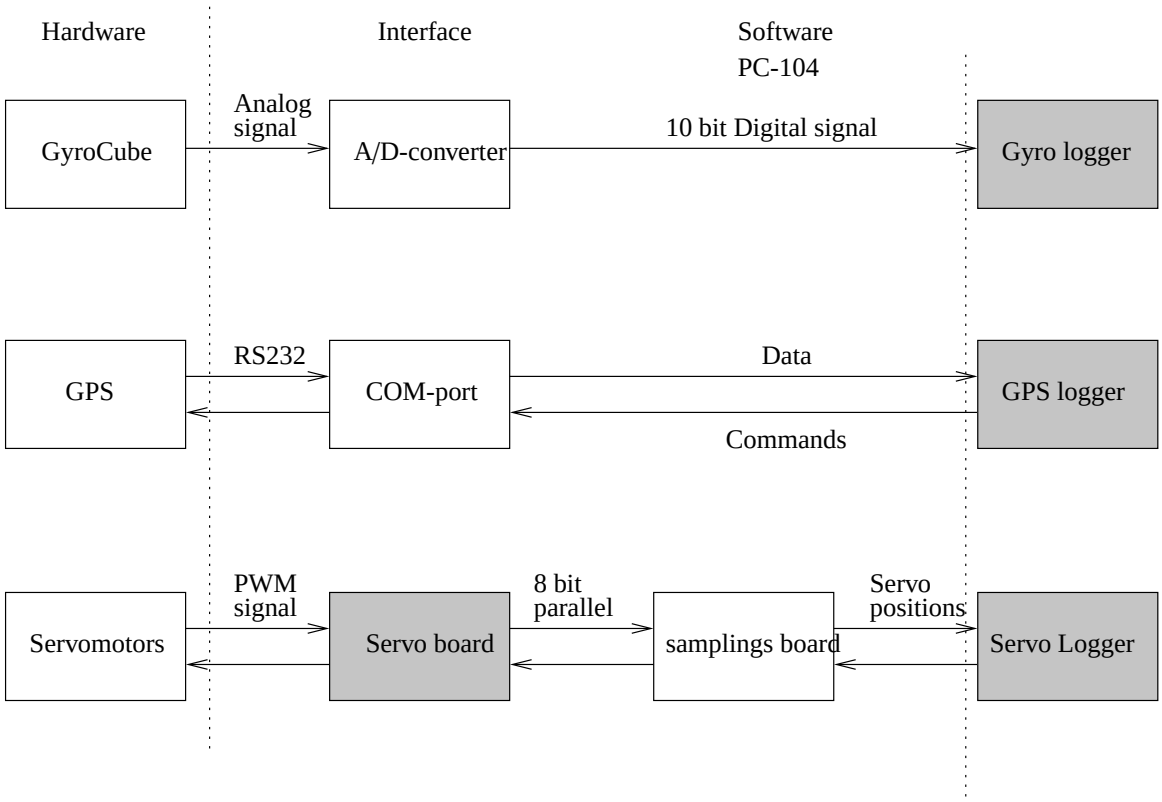


Figure A.1: *The figure show hardware and interface to the gyro logger, GPS logger, and servo logger. The gray boxes are those who require software.*

The servo logger must convert this byte to an deflection angle on either the aileron, rudder, elevator, or flaps. When the aircraft is in autonomous mode is bytes sent to the servo board.

Servo board must both convert PWM signals to bytes and convert bytes to PWM signals. This is done by a PIC18F458 microprocessor. The PIC processor is controlled by the PC-104.

The last section of the chapter is a user guide to the data logger.

A.1 Gyro Logger

The gyro logger is a S-function in Simulink, which sample the GyroCube through the PC-104 sampling board. The output from the GyroCube are ${}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}$, and translatory acceleration ${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}}$. These outputs are an analog signal between 0V and 5V. This is sampled with the 12 bit A/D-converter on the samplings board. The equations used to convert the byte values to angular velocity and accelerations are found in appendix B.

A.2 GPS Logger

The GPS logger logs position and velocity of the aircraft given in $\mathbb{E}$. It also log the number of satellites, in order to indicate if the GPS has contacts to any satellites.

The design of the GPS logger is shown in figure A.2, and it consist of three S-functions which:

recrs232.c This function receive data frames from the GPS module, with a RS232 connection on COM-port 2, and send them to GPSfunc.c.

sendrs232.c This function is used to send data frames to the GPS module using COM-port 2 on the PC-104.

GPSfunc.c The input to this function is data frames from the GPS module. Theses frames are interpreted and output from this function is coordinates and velocities in $\mathbb{E}$. There is also an output, there gives number satellites and another one gives the commands sent to the GPS module.

The communication between the GPS logger and SUPERSTAR II GPS is data frames. The next section describes the most important data frames, used to control the GPS, and get position and velocity.

A.2.1 Input and Output from the GPS

The S-function `gpsfunc.c` controls the GPS and get position and velocity. The function receive data from the GPS and send data frames to the GPS. From those data frames is relevant information like position and velocity saved in the file gpsdata.dat.

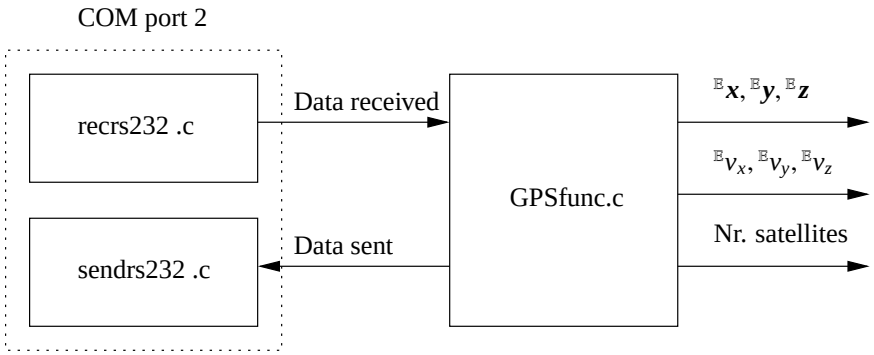


Figure A.2: *The GPS logger is implemented in three S-functions, recrs232.c used to receive data from the GPS, sendrs232.c used to send commandos to the GPS module, and GPSfunc.c which control the GPS and write the data in a file.*

A data frame consists of the data fields given in table A.1. It have a header with four bytes, a data field, and a checksum field. The checksum field is used to control the transmission for errors. The checksum is calculated as the sum of the all the byte in the frame.

Byte Nr	**Name**	**Description**	**Data type**
0	SOF	Start of header, is always 0x01	unsigned char
1	ID	Identify type of message	unsigned char
2	Comp. ID	Control field, and must be 255-ID	unsigned char
3	Lenght	Give length of data.	unsigned char
4-259	Data	Data field	
Length + 4	Checksum	Check for errors, and is the sum all the bytes.	short int

Table A.1: *Description of a data frame*

To set up the gps module and get data, the following frames are used:

Initiate link This frame initiate the link, and when the gps module is finish would respond with the same message, and has number #63.

Set receiver configuration This frame is used to setup the receiver. Here is the sample frequency, type of antenna and information like maximum velocity saturation. It has number #30.

DGPS Configuration Here is it chosen that the SBAS system is chosen instead of using DGPS. This frame has ID number #83

Request navigation data This frame is sent to request ECEF navigation data continuously with a sample rate on 5Hz. This frame has ID number #21. When this frame are sent, will the GPS send a frame with ID number #21 every 0.2sec containing position and velocity of the aircraft.

All information about the different data frames can be found in [11]. Table A.2 shows the most important data fields in the data frame with ID #21. This frame gives position and velocity of the aircraft.

Byte Nr	**Name**	**Description**	**Data type**
15-22	${}^{\mathbb{E}}\boldsymbol{P}_x$	Coordinate in x-direction	double
23-30	${}^{\mathbb{E}}\boldsymbol{P}_y$	Coordinate in y-direction	double
31-38	${}^{\mathbb{E}}\boldsymbol{P}_z$	Coordinate in z-direction	double
39-42	${}^{\mathbb{E}}v_x$	Velocity in x-direction	float
43-46	${}^{\mathbb{E}}v_y$	Velocity in y-direction	float
47-50	${}^{\mathbb{E}}v_z$	Velocity in z-direction	float
80	Nr. Sat	Number of satellites used in solution	unsigned char

Table A.2: *Important fields in the dataframe "Navigation data".*

A.3 Servo Logger

The servo logger has two functions, save position of servomotors when it is in manual mode, and servomotors send positions to the servoboard when it is in autonomous mode. The servo logger communicate with the servo board via the samplings board. This samplings board have a 16 bit digital I/O port, which is used to a parallel data communication. This section describes this protocol designed for this purpose.

The protocol use 11 bit to communicate with the servo board:

Mode is used to detect whether the aircraft is in manual mode when the bit is high, or in autonomous mode when the bit is low. This information is used to decide if the servo logger must send servo positions to the servo board, or log servo positions from the servo board.

Request is controlled of the servo logger, and is active low. The servo logger pull this bit low, when it transfers a byte, and it go high when the data transfer is finish.

PIC ready is controlled by the servo board, and this bit go low when the servo board is ready to respond the servo logger.

Data is a byte representing the positions on the servo motors.

The servo board sample the servo positions with 40Hz. For each sample time is it possible to log data from seven servomotors, or control 5 servo motors. When a sample time start initiate the servo logger the communication by setting Request low, see figure A.3. When the aircraft is in manual mode the servo board puts a byte on the bus, and set PIC-ready low. The servo logger reads the byte and set Request high when it is finish. The servo board detects this, and remove the data from the bus set PIC-ready high.

In autonomous mode the servo logger also initiates the communication by setting Request Low and the servo board answers by setting PIC-ready. The servo logger puts data on the bus and sets Request high. The servo board reads the bus and set PIC ready high when it is finish. The servo logger remove the data from the bus.

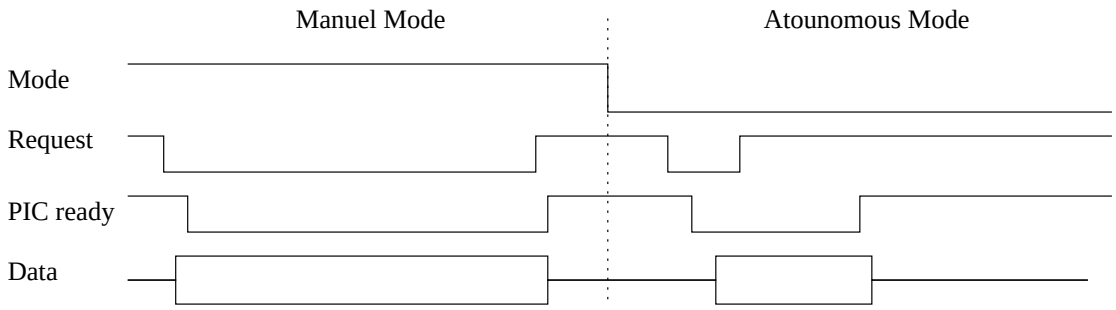


Figure A.3: *The figure show the protocol for the sending a data byte between the servo board and the PC-104 in both manual and autonomous mode.*

The data from the servomotors are saved in the file servodata.dat. In manual mode is servo positions saved, and in autonomous is the control signals for the servomotors saved.

A.4 Servo Board

The input to the servomotors is a PWM signal, and the output to the PC-104 is a byte. It is therefore necessary to make a conversion between bytes and PWM signals. The servoboard has two functions, it must convert PWM signals to bytes and implement the protocol for parallel communication with the PC-104, specified in section A.3. These two functions are implemented in C on a PIC18F458 micro controller. This section describe the reading and generation of PWM signals.

A.4.1 Signal Conversion

The PWM signal for the servomotors are shown on figure A.4. This signal must have a periode smaller than 30ms, else would the servomotors not keep the position. The signal is high between 1ms and 2ms. This time decide the position of the servomotors. The PIC processor must be able to convert both from bytes to PWM signals in autonomous mode, and from PWM signals to bytes in manual mode.

Conversion from PWM Signals to Bytes

The servoboard is designed so it is possible to sample PWM signals from 7 servomotors. The signals go to the PORTB on the PIC. This port detects when the PWM signal change and generate an interrupt. Timer0 in the PIC is then used to measure the time to the port change

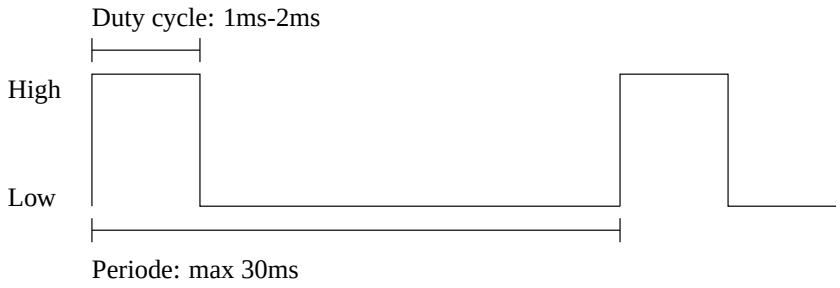


Figure A.4: *The figure show the PWM signal, which is sent to the motor. The high part of the signal is the duty cycle.*

again. This timer run all the time, and every time a the pwm signal change, is the value of Timer0 saved. The principle for measuring a PWM signal can be seen on figure A.5.

Figure A.5: *The figure show when the value of Timer0 is read.*

1. An interrupt are generated when the PWM signal go high, and the value in Timer0 are saved in a variable.

2. A new interrupt are generated when the PWM signal go low, and the value of Timer0 are saved. This value is subtracted from the value in (1), and the time where the signal is high are found. Then 1ms are subtracted from the time and the position of the servomotor are found.

3. When the signal goo high again, the time is calculated. This time represent the time the PWM signal are low, and this time is discarded.

Conversion from Bytes to PWM Signals

When the aircraft is in autonomous mode it is necessary to generate PWM signals to control the servomotors. In this mode is the PC-104 sending a new position for each of the five servomotors with an sample frequency of 40Hz. When the position of the servomotors are received, are the following operations are explain made:

1. The five servo bytes representing servo positions are saved in an array.

Data from PC-104	
Servomotor nr	Value
1	0xa0
2	0x00
3	0xff
4	0xa0
5	0xa1

Sorting data	
Servomotor nr	Value
2	0x00
1	0xa0
4	0xa0
5	0xa1
3	0xff

Calculate PWM-table	
Servomotor nr	Value
start	constant
2	0x00
1	0xa0
4	0x00
5	0x01
3	0x5e

Figure A.6: *The figure shows how the data from the PC-104 is handled. The data is first sorted, and then the difference is calculated between the signals.*

2. The positions are sorted so the lowest bytes first, and the highest byte last.

3. Then the difference between servomotor n and n+1 is found. This value are saved in a new array PWM table. The first value in this array is a constant and represent the first ms of the PWM signal.

Two timers are used generate the PWM signal. Timer1 are used to control the periode time. This timer generate an interrupt for each 20ms. When this interrupt occur(see figure A.7) are all PWM signal for the motors set high. Now timer2 is used to control, when the signals for the different servomotors should go low, and it use the PWM array for this. Figure A.7 shows in what order the servomotors goes low, after the values specified in figure A.6.

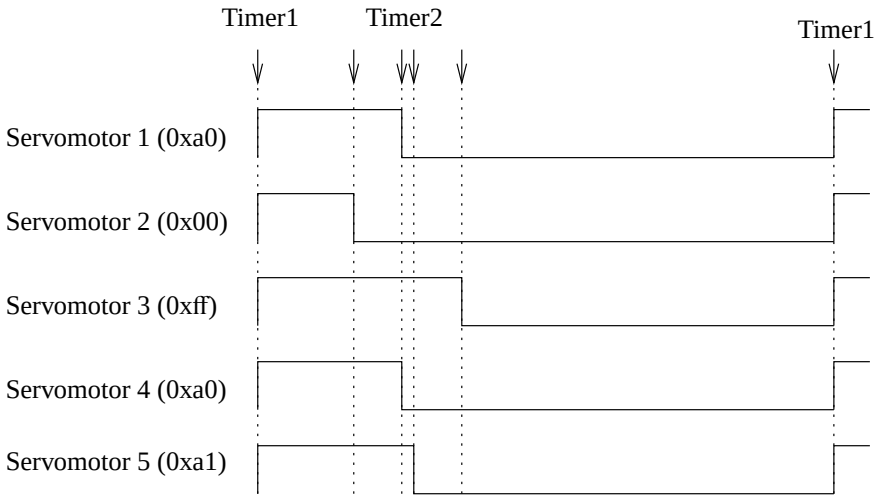


Figure A.7: *It is shown how Timer1 and Timer2 is used to generate the PWM signal.*

A.5 Guide to Data Logger

The data logger requires that Matlab XPC is properly installed on a PC, for further instruction on this subject see [9]. The XPC toolbox can be tested by running the command xpctest, when the PC-104 is running and connected to the host. If this test is a success, is the PC configured correct and it is possible to run the data logger.

The following steps is done in order to use the data logger:

1. The first time the test logger is executed, is it necessary to compile the following files using the command *mex filename.*
   - adcblock.c
   - gpsfunc.c
   - recrs232.c
   - sendrs232.c
   - servoboard1.c
2. Open the file testflight.mdl in the XPC library, and turn on the PC-104. Connect the host PC with the PC-104 using a RS232 null modem cable, using COM-port 1 on the host.
3. Build testflight.mdl with the command Ctrl+b, while in simulink.
4. When this is done, is first the Graupner MX transmitter turned on, then the servo board. The data logger is executed with the command *tg.start*. It is now possible to remove the RS-232 cable and make a test flight. Whit the current settings is it possible to save data in 30min.
5. After the test flight is the PC-104 connected to the Host and can be stopped with the command *tg.stop*.
6. Files containing the sensor data can be transferred to the host using the script getfiles.m. All data is saved in two files, a high resolution file containing all samples, and and a low resolution file containing every 10. sample. Getfiles.m get the files which only sample every 10. sample. This is used to control the data after a test flight. This is done because the data transfer is over the RS232 connection and it would take very long time to get all the data. The high resolution data can be read by connecting the harddisk to a desktop computer or run the command dontuse.

Appendix B

Test Appendix

This appendix contains the equations used to convert the output data from the sensors to units compatible with the flight simulator. This is necessary for data from the GyroCube and the servo board.

The output from the GyroCube and the servo board cannot be send directly to the simulator. The output is an analogue signal converted by the A/D-converter on the samplings board. This gives a 12 bit number, which must be converted to $^{rad}/_{s}$ for the gyros and $^{m}/_{s^2}$ for the accelerometers. The servo board samples the position of the servos and represents this position as a byte. This byte is converted to an deflection angle in radians for flaps, aileron, elevator, and rudder.

Calibration of GyroCube

The output from the gyroscope is an analog signal between 0V and 5V, which are sampled by the A/D samplings board. It is necessary make a conversion of the data from the A/D-converter.

B.0.1 Gyro

In order to convert the output from the gyro is a test performed where the GyroCube is rotated $\frac{\pi}{2}$rad several times around one axis. The gyroscope measures the angular velocity and this output is integrated so the output are the position of the gyroscope. It is then possible to find the factor, the output signal must be multiplied with, to convert the signal s to radians.

The output from the GyroCube is an analog signal between 0V and 5V, where 2,5V are $0\frac{rad}{s}$. This signal is sampled with the samplings board and (B.1) describes how a bit value is converted to a voltage with a bias on 2,5V. C_s is found by looking at the solution and working area of the A/D-converter on the samplings board. The samplings board has a 12 bit solution in the range -5V to 5V, so C_{bit} become $\frac{10\text{V}}{2^{12}\text{bit}} = 0,0024\frac{\text{V}}{\text{bit}}$. $bias_{digital}$ is $\frac{3}{4}$ of the solution on the A/D-converter and $bias_{digital}$ become 3072.

$$\begin{aligned} V_G &= C_s bit_{value} + bias_{digital} \\ &= 0.0024(bit_{value} + 3072) \end{aligned} \tag{B.1}$$

After using equation (B.1) is the byte value converted to a voltage. Test of the gyros show that when the GyroCube don't rotate, there is a bias from 2,5V, and it is not the same for all three gyros. This bias is found from the data logged in the test flight. A mean is taken over a periode where the aircraft is standing still. This mean value is the bias of the signal. The bias for the angular velocity is.

$$^{\mathbb{B}}\omega_{x,bias} = 0.0377 \tag{B.2}$$

$$^{\mathbb{B}}\omega_{y,bias} = -0.0584 \tag{B.3}$$

$$^{\mathbb{B}}\omega_{z,bias} = -0.0584 \tag{B.4}$$

It is necessary to find the coefficient which convert the voltage to $\frac{rad}{s}$. An experiments is made where the GyroCube is rotated 90 ° several times with different angular velocity. This signal is then integrated, and the outcome is the position of the gyro. It is then possible to find the factor which must be multiplied on the signal. The result of this test is shown in figure B.1. The top graph shows the voltage of the signal. The test shows that the signal must be multiplied with 3.35 in order to get $\frac{\pi}{2}$rad.

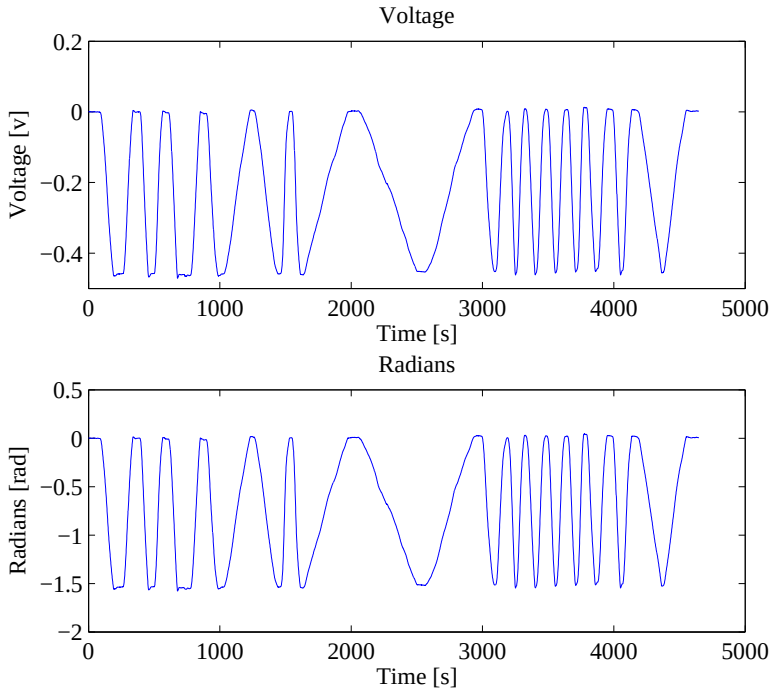


Figure B.1: *The top graph show the input voltage integrated. The other graph show the position of the GyroCube.*

The equation to convert from a byte value to $\frac{rad}{s}$ is given in (B.5) to (B.7). ${}^{\mathbb{B}}\boldsymbol{\omega}_x$, and ${}^{\mathbb{B}}\boldsymbol{\omega}_y$ is multiplied with -1, so the fit with the direction of $\mathbb{B}$.

$$
\begin{aligned}
{}^{\mathbb{B}}\omega_x &= -\Big(\big(C_s bit_{value} + bias_{digital}\big) + {}^{\mathbb{B}}\omega_{x,bias}\Big)\,3.35 && \text{(B.5)}\\
{}^{\mathbb{B}}\omega_y &= -\Big(\big(C_s bit_{value} + bias_{digital}\big) + {}^{\mathbb{B}}\omega_{y,bias}\Big)\,3.35 && \text{(B.6)}\\
{}^{\mathbb{B}}\omega_z &= \Big(\big(C_s bit_{value} + bias_{digital}\big) + {}^{\mathbb{B}}\omega_{z,bias}\Big)\,3.35 && \text{(B.7)}
\end{aligned}
$$

B.0.2 Accelerometer

This section contains the conversion for the accelerometers. As for the gyros, is the input a 12 bit number from the A/D-converter. This number can also be converted to a voltage using equation (B.1). Figure B.2 is an example of the output from the accelerometers. The top graph shows the output in volts. ${}^{\mathbb{B}}a_{\mathbb{B},x}$ and ${}^{\mathbb{B}}a_{\mathbb{B},y}$ measures no acceleration in the start and in the end. ${}^{\mathbb{B}}a_{\mathbb{B},z}$ measure the gravity. There is a bias and this can be found by taking the mean of signal in the time where the aircraft don't move. The bias is given as:

$$a_{bias} = 0.8363 \tag{B.8}$$

The difference between ${}^{\mathbb{B}}a_{\mathbb{B},z}$ and the two other components is 1v, or 1g. A voltage can be converted from voltage to acceleration by multiplying the signal with $9.82\frac{m}{s^2}$. The bottom graph of figure B.2 shows the signal converted to acceleration.

The equation for converting the output from the accelerometers are:

$${}^{\mathbb{B}}\boldsymbol{a}_{\mathbb{B}} = -((C_s bit_{value} + bias_{digital}) - a_{bias}) \cdot 9.82 \tag{B.9}$$

Servo Motor Conversion

The servo board samples the servo motors and sends this value to the PC-104. These values represents deflection angles on flaps, aileron, elevator, and rudder. This section describes how the byte values are converted to deflection angles.

A test is performed on the aileron, where a position is sent to the servo motor via the transmitter. The angle of the aileron is measured and the position of the servo motor is logged. This is done for many different angles, and the result can be seen on figure B.2.

The straight line which gives the best fit is given in (B.10). The equation have the opposite sign of the graph, in order to have the right angle with respect to $\mathbb{B}$.

$$\delta_a = -(0.0068 byte_a - 0.6761) \tag{B.10}$$

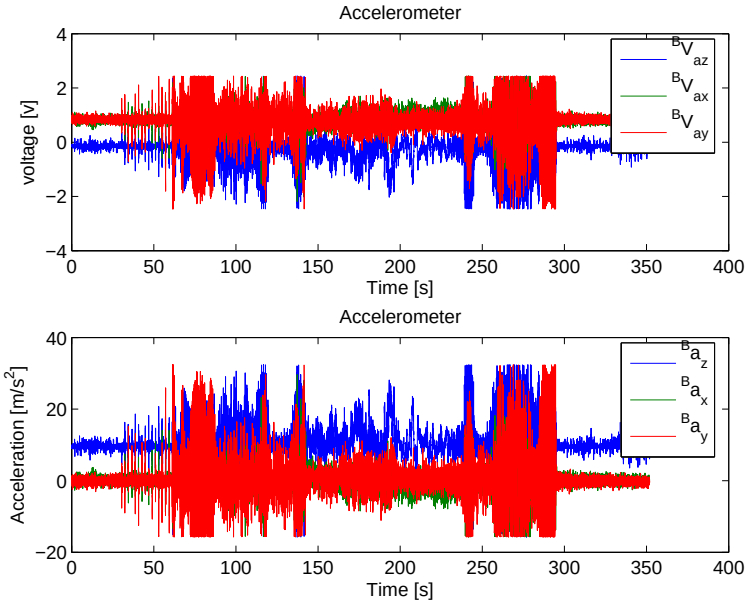


Figure B.2: *The figure show the output from the accelerometer in voltage in the top graph, and in acceleration in the bottom graph.*

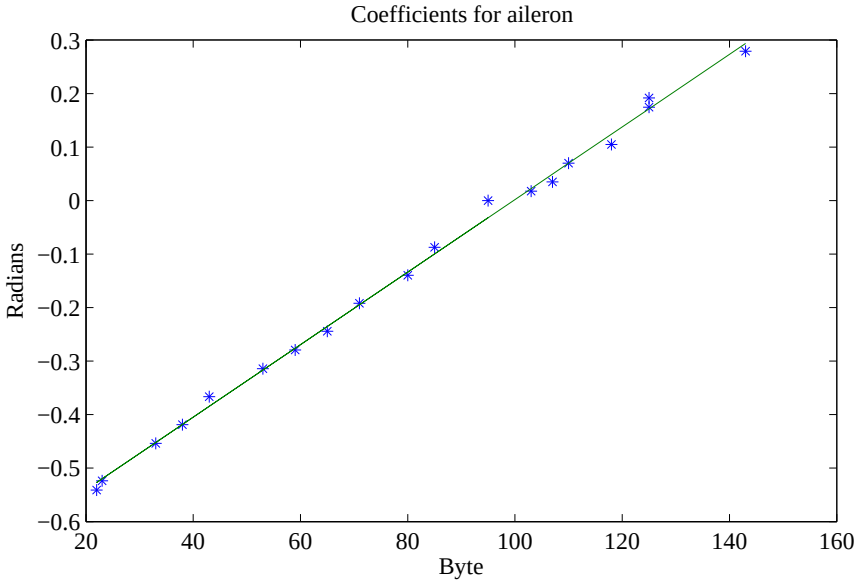


Figure B.3: *The figure shows the angle of the aileron as a function of the byte value.*

The deflection for the other deflection angles are found by using the same method, and the functions are given as:

$$\delta_f = -(0.3977byte_f + 65.6205)\frac{\pi}{180} \quad \text{(B.11)}$$

$$\delta_e = (0.3184byte_e - 30.248)\frac{\pi}{180} \quad \text{(B.12)}$$

$$\delta_r = (0.2246byte_r - 18.668)\frac{\pi}{180} \quad \text{(B.13)}$$

Appendix C

Quaternions

In this appendix is a description on how to use Quaternions to represent rotations of different coordinate systems or reference frames.
This appendix starts with a description of the quaternion and some fundamental properties of the quaternion together with an example of the use of the quaternions. Finaly is the time derivative of a quaternion derived, which is used in this project to represent the rotational kinematic model of the aircraft.

Definitions

A quaternion is a hypercomplex number, which is an extension to the complex numbers. It is represented similar to complex numbers with a real part and imaginary parts. A quaternion (C.1) has four parameters called Euler parameters, three imaginary q_1, q_2, q_3 and one real q_4 defined in (C.1), representing a rotation of θ about an axis $\hat{\boldsymbol{e}}$,

$$\begin{aligned} \boldsymbol{q} &= q_4 + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 \\ &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \hat{\boldsymbol{e}} \sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix} \end{aligned} \tag{C.1}$$

where $\hat{\boldsymbol{e}} = \begin{bmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \end{bmatrix}^{\mathrm{T}}$ is the unit vector representing Euler's eigenaxis[1] and θ is the rotation angle. The imaginary numbers $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are defined by the fundamental formula of quaternions (C.2) discovered by William Rowan Hamilton [1].

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1 \tag{C.2}$$

Since the quaternion is a hypercomplex number the conjugated quaternion is defined as (C.3),[1]

$$\boldsymbol{q}^* = q_4 - \boldsymbol{i}q_1 - \boldsymbol{j}q_2 - \boldsymbol{k}q_3 \tag{C.3}$$

[1] Euler's eigenaxis is the axis which an object is rotated about [1].

One important property of quaternions are that they have the norm of 1 thus (C.4) is true [1].

$$|\boldsymbol{q}| \triangleq \sqrt{\boldsymbol{q} \otimes \boldsymbol{q}^*} = \sqrt{\boldsymbol{q}^* \otimes \boldsymbol{q}} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \equiv 1 \tag{C.4}$$

where $\otimes$ denotes that quaternion multiplication is used see section C and the $\boldsymbol{q}^*$ is the conjugated quaternion. The norm (C.4) especially comes in handy in control application with regards to stability, because it bounds the size of the quaternion.

From (C.4) it can be derived that the inverse quaternion is defined as the complex conjugated quaternion thus giving (C.5)

$$\forall \boldsymbol{q}(|\boldsymbol{q}| = 1)|\boldsymbol{q}^{-1} \equiv \boldsymbol{q}^* \tag{C.5}$$

Quaternion Multiplication

This section contains the rules of multiplying two quaternions and an example of the usage of quaternion multiplication.

The quaternions are noncommutative and associative, thus usual multiplication is not possible. Instead it has to be carried out by quaternion multiplication (C.6) sometimes denoted by $\otimes$ or understod in the context of the equations[1].

$$\begin{aligned}
\boldsymbol{q}_C &= \boldsymbol{q}_A \boldsymbol{q}_B = \boldsymbol{q}_A \otimes \boldsymbol{q}_B \\
&= \begin{bmatrix} \boldsymbol{q}_{A_{1:3}} q_{B_4} + q_{A_4} \boldsymbol{q}_{B_{1:3}} + \boldsymbol{q}_{A_{1:3}} \times \boldsymbol{q}_{B_{1:3}} \\ q_{A_4} q_{B_4} - \boldsymbol{q}_{A_{1:3}}^{\mathrm{T}} \boldsymbol{q}_{B_{1:3}} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{S}(\boldsymbol{q}_{A_{1:3}}) + q_{A_4} \boldsymbol{I}_{3\times 3} & \boldsymbol{q}_{A_{1:3}} \\ -\boldsymbol{q}_{A_{1:3}}^{\mathrm{T}} & q_{A_4} \end{bmatrix} \boldsymbol{q}_B \qquad \text{(C.6)} \\
&= \begin{bmatrix} -\boldsymbol{S}(\boldsymbol{q}_{B_{1:3}}) + q_{B_4} \boldsymbol{I}_{3\times 3} & \boldsymbol{q}_{B_{1:3}} \\ -\boldsymbol{q}_{B_{1:3}}^{\mathrm{T}} & q_{B_4} \end{bmatrix} \boldsymbol{q}_A \qquad \text{(C.7)}
\end{aligned}$$

Where $\boldsymbol{S}(\boldsymbol{a})$ is a skew-symmetric matrix (C.9) of a vector $\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^{\mathrm{T}}$ defined as (C.9) and found through (C.8):

$$\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{S}(a)\boldsymbol{b} = -\boldsymbol{S}(b)\boldsymbol{a} \tag{C.8}$$

$$\boldsymbol{S}(\boldsymbol{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{C.9}$$

Another property of the quaternion is that it is a division algebra, thus (C.10) is true:

$$\boldsymbol{q}_C = \boldsymbol{q}_A \boldsymbol{q}_B \Leftrightarrow \boldsymbol{q}_B = \boldsymbol{q}_A^{-1} \boldsymbol{q}_C \tag{C.10}$$

Rotation by Quaternions

As described above a quaternion $\boldsymbol{q}$ represents a rotation in $\Re^3$, thus rotating a vector between two viewpoints or coordinate systems is done as in the following example.

Given two coordinate systems $\mathbb{R}$ and $\mathbb{B}$ and a vector ${}^{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}}$ given in $\mathbb{R}$ representing the angular velocity of $\mathbb{B}$. To find the same vector given in $\mathbb{B}$ it is necessary to rotate the vector from $\mathbb{R}$ to $\mathbb{B}$. If the quaternion ${}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}$ represents the rotation from $\mathbb{R}$ to $\mathbb{B}$ then it is possible to rotate the vector ${}^{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}}$ to $\mathbb{B}$ by (C.11) and back to $\mathbb{R}$ by (C.12).

$$\begin{bmatrix} {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} = {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q} \otimes \begin{bmatrix} {}^{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}^{*} \tag{C.11}$$

$$\begin{bmatrix} {}^{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} = {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}^{*} \otimes \begin{bmatrix} {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q} \tag{C.12}$$

Where $\begin{bmatrix} {}^{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix}$ is the original vector put on a quaternion form with the real part $q_4 = 0$, note that this is not a true quaternion since it doesn't have to have a norm of 1.

If another coordinate system $\mathbb{E}$ seen from $\mathbb{B}$ is rotated by ${}^{\mathbb{E}}_{\mathbb{B}}\boldsymbol{q}$, then ${}^{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}}$ seen from $\mathbb{E}$ is given as:

$$\begin{bmatrix} {}^{\mathbb{E}}_{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} = {}^{\mathbb{E}}_{\mathbb{B}}\boldsymbol{q} \otimes {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q} \otimes \begin{bmatrix} {}^{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ 0 \end{bmatrix} \otimes {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}^{*} \otimes {}^{\mathbb{E}}_{\mathbb{B}}\boldsymbol{q}^{*} \tag{C.13}$$

Thus showing that shifting between different coordinate systems is straight forward by using quaternion multiplication.

The Time Derivative of a Quaternion

This section contains the derivation of the time derivative of a quaternion. Which is usefull when predicting how object rotates or have rotated.

The method used is to solve (C.14) for a quaternion $\boldsymbol{q}(t)$ changing over time.

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{f(t + \Delta t) - f(t)}{\Delta t} \tag{C.14}$$

Let $\boldsymbol{q}(t)$ be a quaternion changing continuously over time and let $\boldsymbol{q}(\Delta t)$ be a rotation over a very small amount of time, thus a very small rotation from this we get:

$$\boldsymbol{q}(t + \Delta t) = \boldsymbol{q}(t) \otimes \boldsymbol{q}(\Delta t) \tag{C.15}$$

Joining (C.14) with (C.15) gives:

$$\frac{\mathrm{d}\boldsymbol{q}(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{\boldsymbol{q}(t + \Delta t) - \boldsymbol{q}(t)}{\Delta t} \tag{C.16}$$

Looking closer to the $\boldsymbol{q}(\Delta t)$ and the approximation, that for a very small amount of time the rotational angle is very small and $\lim_{\Delta\theta \to 0} e\,sin(\Delta\theta) = \Delta\theta$, where $e \leq 1$ thus giving:

$$\boldsymbol{q}(\Delta t) = \begin{bmatrix} \hat{\boldsymbol{e}} \sin(\frac{\Delta\theta}{2}) \\ \cos(\frac{\Delta\theta}{2}) \end{bmatrix} \approx \begin{bmatrix} \frac{\Delta\boldsymbol{\theta}}{2} \\ 1 \end{bmatrix} \tag{C.17}$$

Combining (C.17), (C.16) and (C.15) and rewriting using $1 = \frac{\Delta t}{\Delta t}$ and $\boldsymbol{\omega} = \frac{\Delta \theta}{\Delta t}$ gives:

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{q}(t)}{\mathrm{d}t} &= \lim_{\Delta t \to 0} \frac{\boldsymbol{q}(t+\Delta t) - \boldsymbol{q}(t)}{\Delta t} \\
&\approx \lim_{\Delta t \to 0} \frac{\begin{bmatrix} \frac{\boldsymbol{\Delta\theta}\Delta t}{2\Delta t} \\ 1 \end{bmatrix} \otimes \boldsymbol{q}(t) - \boldsymbol{q}(t)}{\Delta t} \\
&\approx \lim_{\Delta t \to 0} \frac{\begin{bmatrix} \frac{\boldsymbol{\omega}\Delta t}{2} \\ 1 \end{bmatrix} \otimes \boldsymbol{q}(t) - \boldsymbol{q}(t)}{\Delta t}
\end{aligned} \tag{C.18}$$

Carrying out the quaternion multiplication (C.7) in (C.18) gives the final representation of the time derivative of a quaternion:

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{q}(t)}{\mathrm{d}t} &= \lim_{\Delta t \to 0} \frac{\begin{bmatrix} -\boldsymbol{S}(\frac{\boldsymbol{\omega}\Delta t}{2}) + \boldsymbol{I}_{3\times 3} & \frac{\boldsymbol{\omega}\Delta t}{2} \\ -\frac{\boldsymbol{\omega}^{\mathrm{T}}\Delta t}{2} & 1 \end{bmatrix} \boldsymbol{q}(t) - \boldsymbol{I}_{4\times 4}\boldsymbol{q}(t)}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\left(\frac{\Delta t}{2} \begin{bmatrix} -\boldsymbol{S}(\boldsymbol{\omega}) & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathrm{T}} & 0 \end{bmatrix} + \boldsymbol{I}_{4\times 4} \right) \boldsymbol{q}(t) - \boldsymbol{I}_{4\times 4}\boldsymbol{q}(t)}{\Delta t} \\
&= \frac{1}{2} \begin{bmatrix} -\boldsymbol{S}(\boldsymbol{\omega}) & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathrm{T}} & 0 \end{bmatrix} \boldsymbol{q}(t)
\end{aligned} \tag{C.19}$$

Where $\boldsymbol{S}(\boldsymbol{\omega})$ is the skew-symetric matrix (C.9) and $\boldsymbol{\omega}$ is the angular velocity vector of which the object observed is rotating to a reference frame given in the rotating frame.

So if we have a coordinate system $\mathbb{B}$ and $\mathbb{R}$ then the change in rotation from $\mathbb{R}$ to $\mathbb{B}$ between the two coordinate systems is given as:

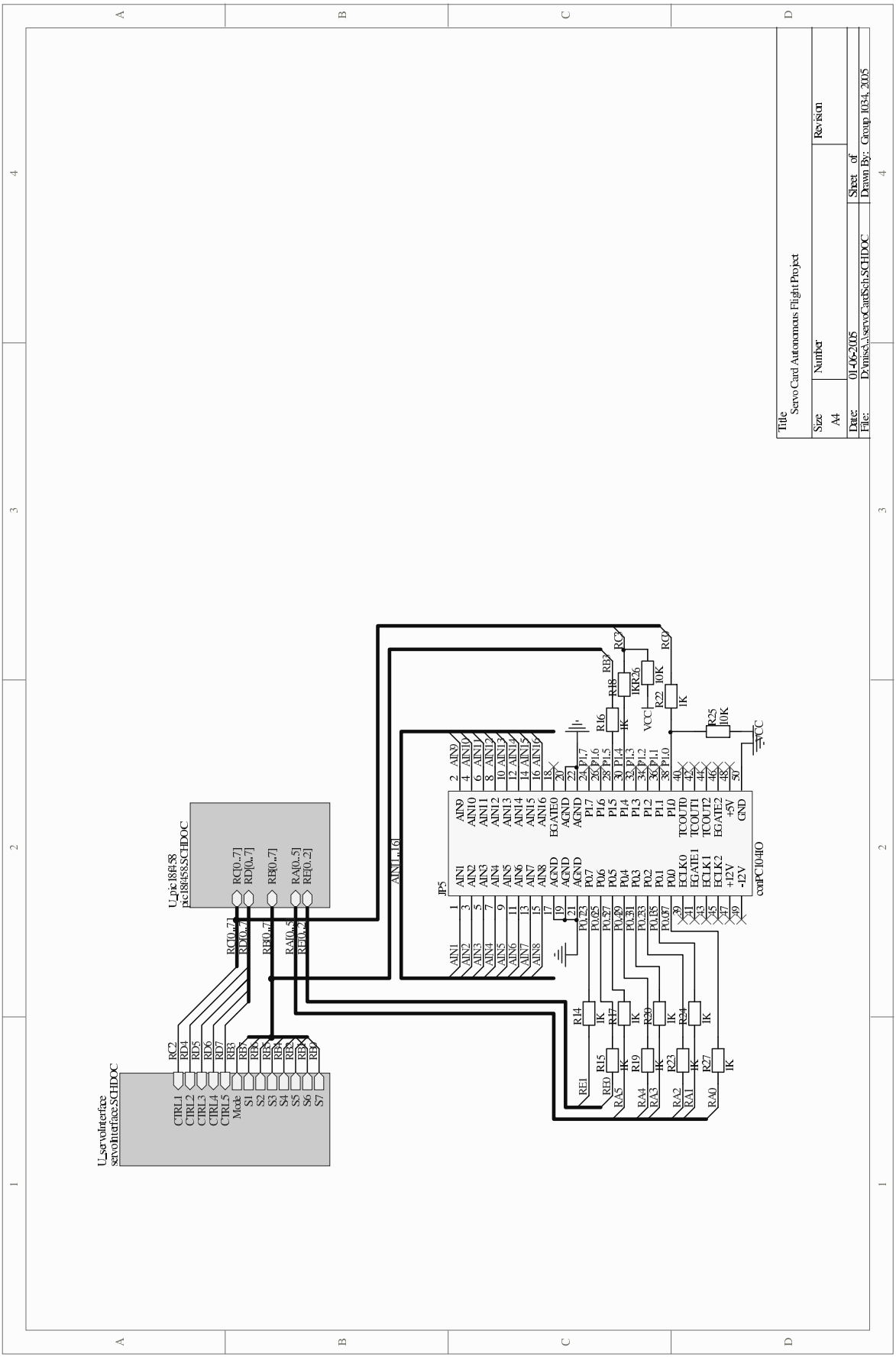
$$\frac{\mathrm{d}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}(t)}{\mathrm{d}t} = \frac{1}{2} \begin{bmatrix} -\boldsymbol{S}({}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}}) & {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{\omega}_{\mathbb{B}} \\ -{}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{\omega}^{\mathrm{T}}_{\mathbb{B}} & 0 \end{bmatrix} {}^{\mathbb{B}}_{\mathbb{R}}\boldsymbol{q}(t) \tag{C.20}$$

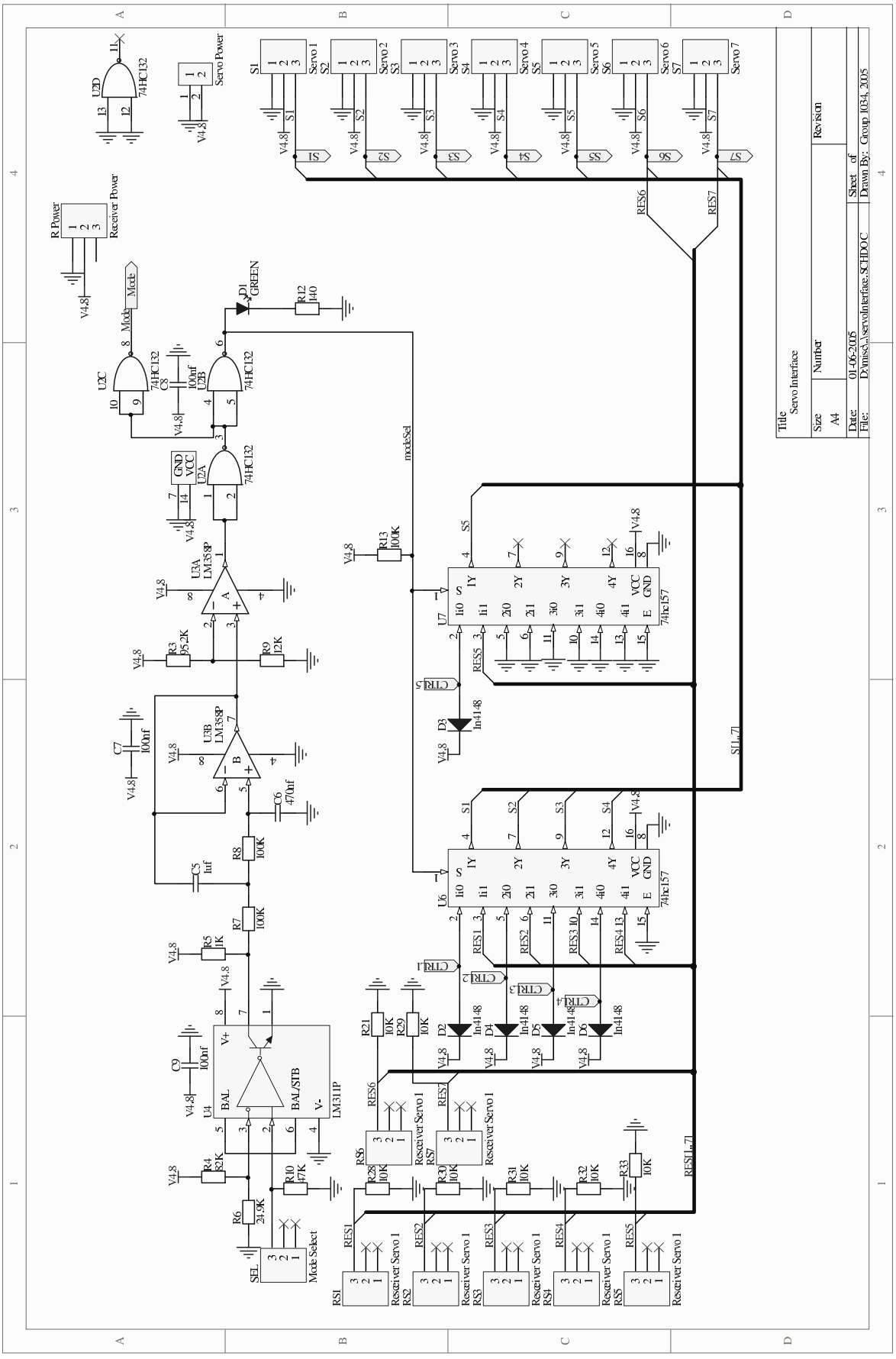
Appendix D

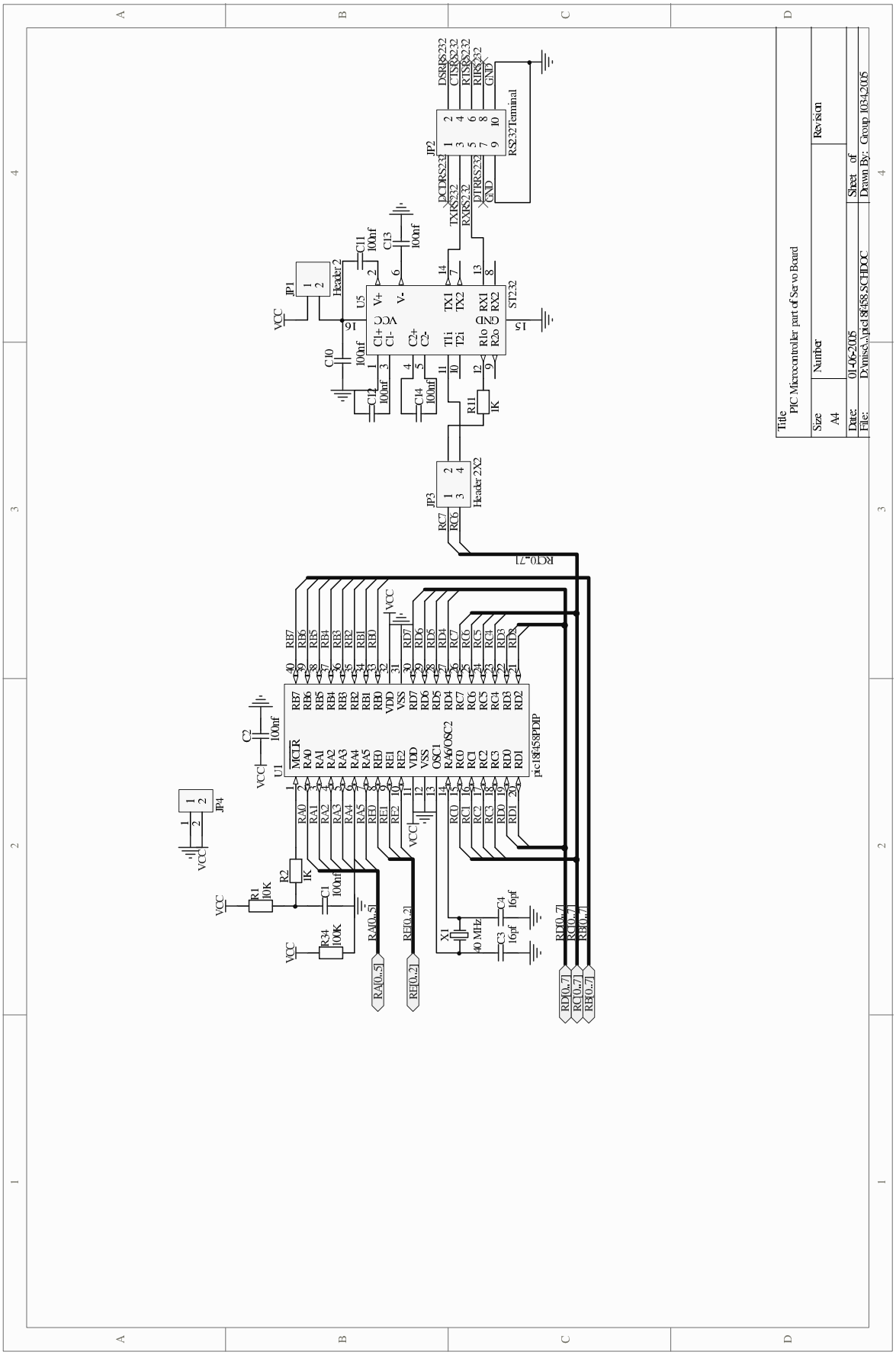
Hardware Implementation

In this appendix are the following schematics included:

- Servo Controller and Interface Board.
  - Overview schematic see page 100.
  - Servo Interface Schematic see page 101.
  - PIC 18f458 Micro controller schematic see page 102.
  - Bill Of Material (BOM) for the Servo Board see page 103.
- PC104 Externsion Card
  - Overview of the PC104 extension card see page 104.
  - Switch mode power supply design see page 105.
  - Gyro Cube 3F interface schematic see page 106.
  - Superstar II GPS interface schematic see page 107.
  - Bill Of Material (BOM) for the PC104 extension card see page 108.

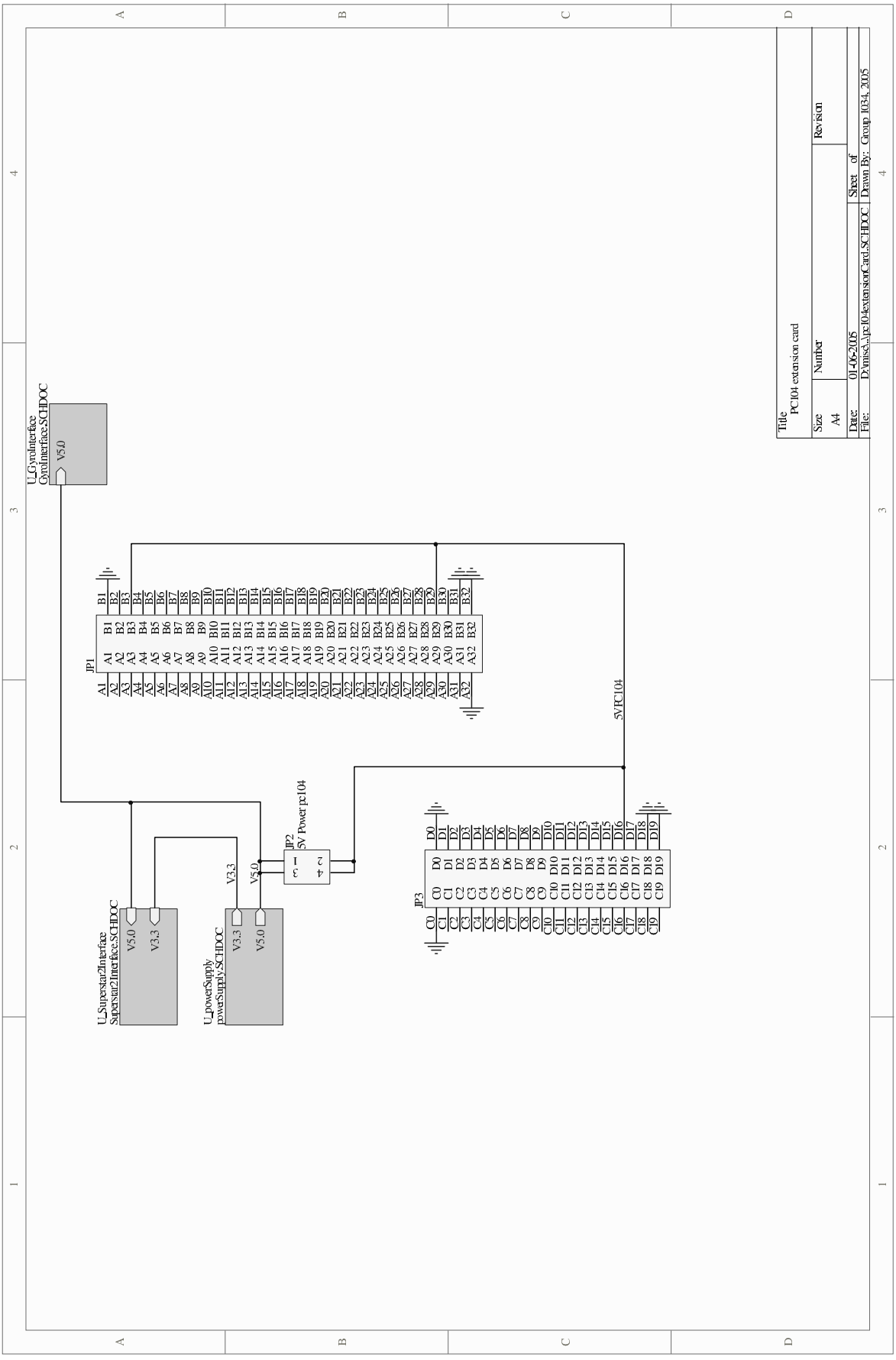

Title
Servo Card Autonomous Flight Project
Size
A4
Number
Revision
Date: 01-06-2005
File: D:\mies\..\servoCardSch.SCHDOC
Sheet of
Drawn By: Group 1034, 2005
U_pic18f458
pic18f458.SCHDOC
RC[0..7]
RD[0..7]
RB[0..7]
RA[0..5]
RE[0..2]
U_servoInterface
servoInterface.SCHDOC
CTRL1
CTRL2
CTRL3
CTRL4
CTRL5
Mode
S1
S2
S3
S4
S5
S6
S7
AIN[1..16]
JP5
conPCI104IO
VCC
R16
R17
R18
R19
R22
R23
R24
R25
R26
R27
10K
1K

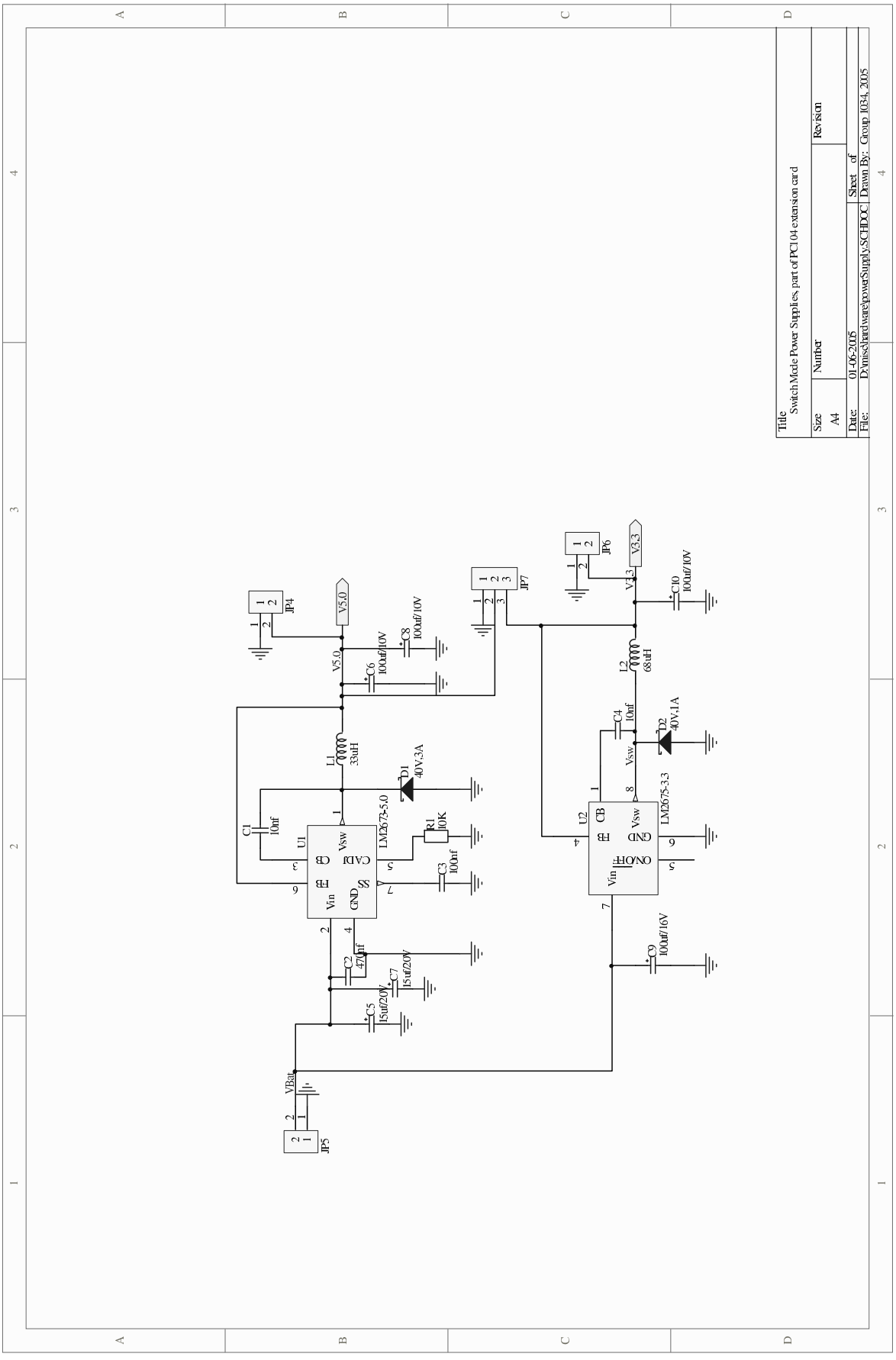

Servo Interface
Size A4
Number
Revision
Date: 01-06-2005
Sheet of
File: D:\mis6\...servointerface.SCHDOC
Drawn By: Group 1034, 2005
Servo 1
Servo 2
Servo 3
Servo 4
Servo 5
Servo 6
Servo 7
Servo Power
R Power
Receiver Power
Mode
Mode Select
Receiver Servo 1
74HC132
LM358P
LM311P
74hc157
In4148
modeSel
RES1
RES2
RES3
RES4
RES5
RES6
RES7
RES[1..7]
S[1..7]
CTR[1..5]

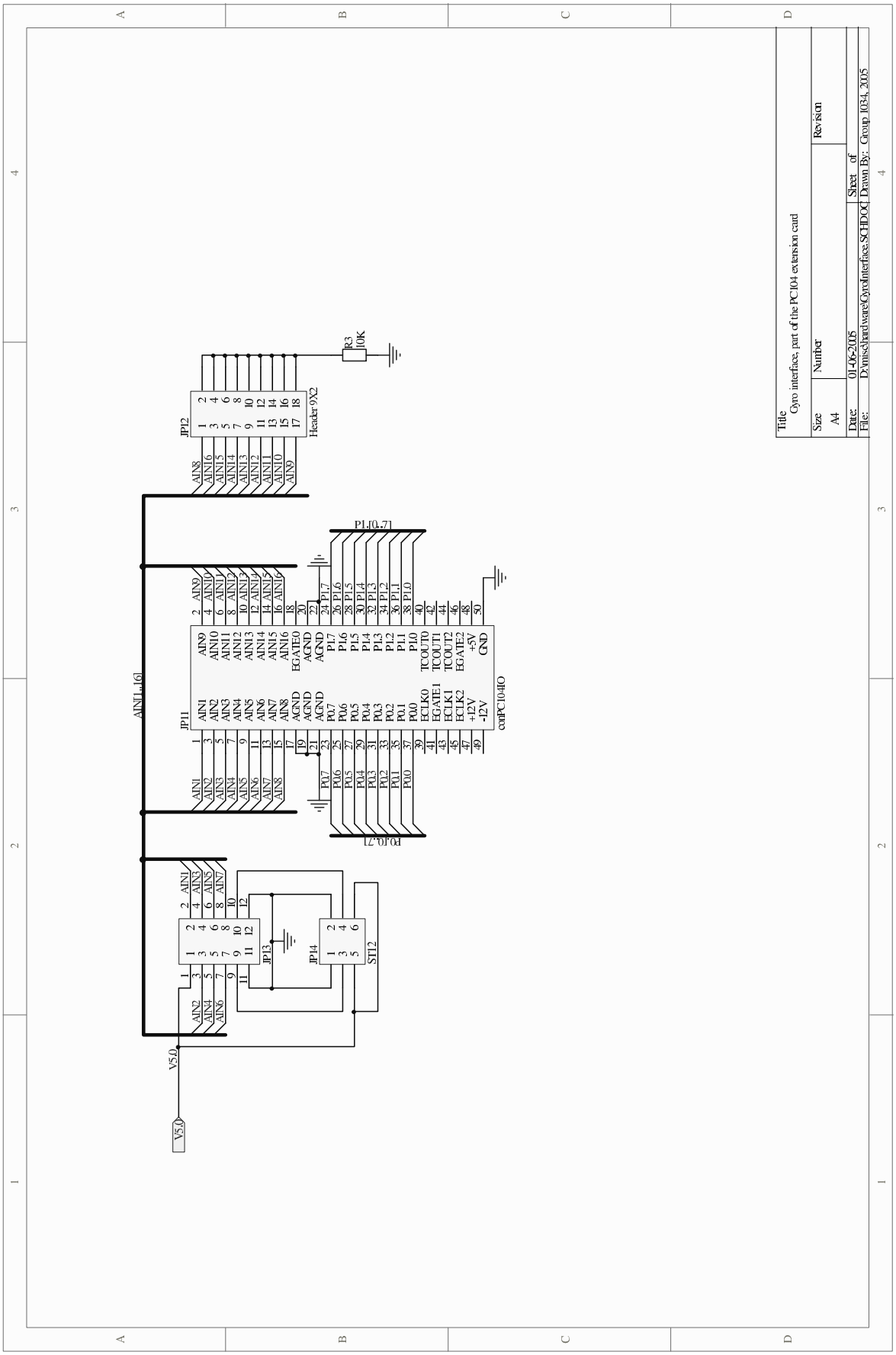

Title
PIC Microcontroller part of Servo Board
Size
A4
Number
Revision
Date: 01-06-2005
File: D:\miss\...\pic18f458.SCHDOC
Sheet of
Drawn By: Group 1034-2005
pic18f458PDIP
RS232Terminal
Header 2
Header 2X2
MAX232
40 MHz
RA[0..5]
RE[0..2]
RD[0..7]
RC[0..7]
RB[0..7]
RC[0..7]

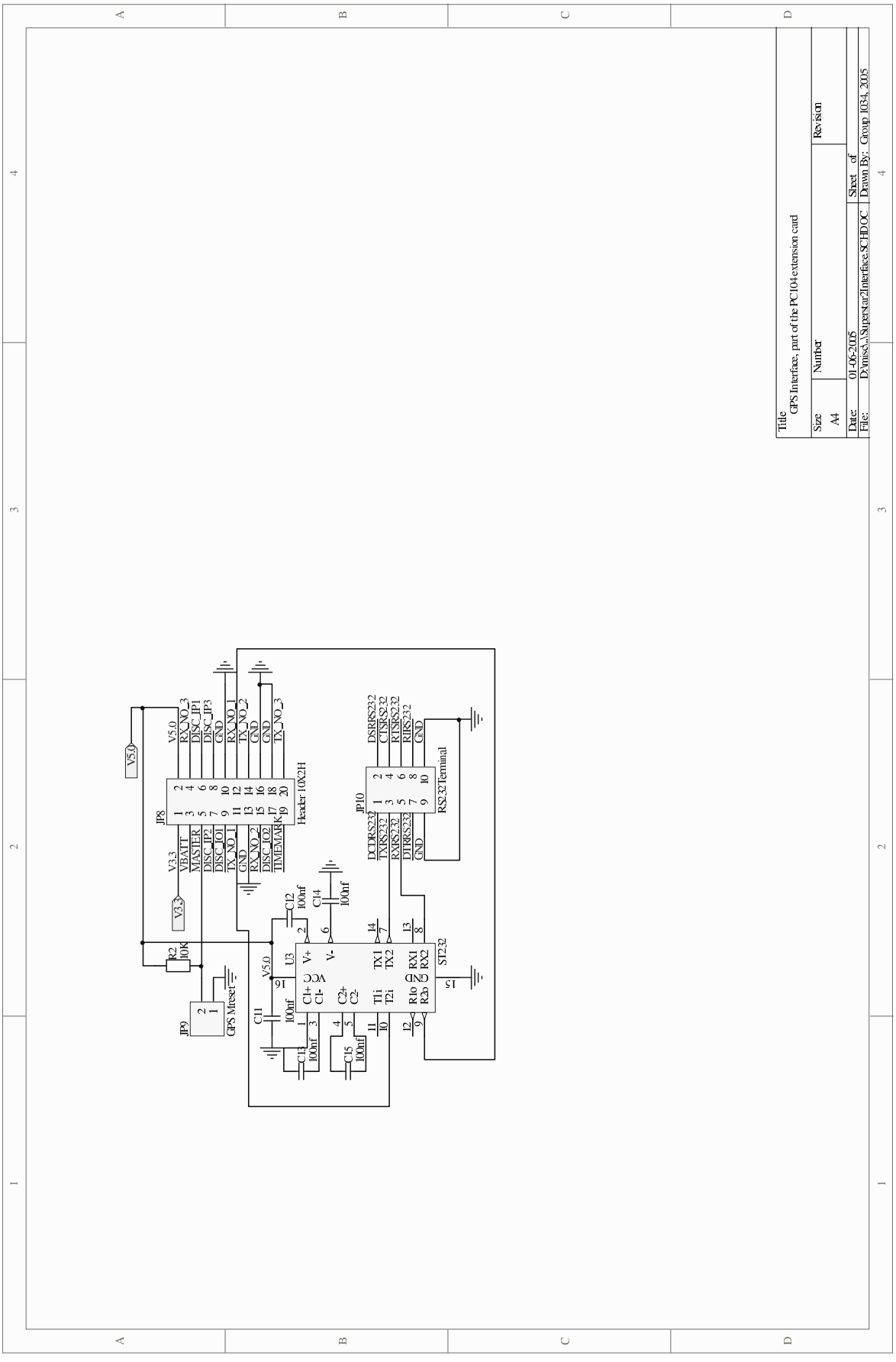
ServoCard: Bill Of Material Group 1034, 2005

Description	Designator	Footprint	Quantity	Value	Farnell code
ceramic capacitor	C1, C2, C7, C8, C9, C10, C11, C12, C13, C14	0805	10	100nf	
ceramic capacitor	C3, C4	0805	2	16pf	
ceramic capacitor	C5	0805	1	1uf	
ceramic capacitor	C6	0805	1	470nf	
Light Emitting Diode (LED)	D1	0805Diode	1		
Diode	D2, D3, D4, D5, D6	0805Diode	5	40V,1A	
Header, 2-Pin	JP1	HDR1X2	1		
Header, 2x5-Pin, Dual row	JP2	HDR2X5	1		
Header, 2-Pin, Dual row	JP3	HDR2X2	1		
2pinVertCon (GUL)	JP4, Servo Power	conVert2pinRobust	2		
Dual row 2x25 pin connector	JP5	con2x25HA	1		
Resistor	R1, R21, R25, R26, R28, R29, R30, R31, R32, R33	0805	10	10K	
Resistor	R2, R5, R11, R14, R15, R16, R17, R18, R19, R20, R22, R23, R24, R27	0805	14	1K	
Resistor	R3	0805	1	95.2K	
Resistor	R4	0805	1	82K	
Resistor	R6	0805	1	24.9K	
Resistor	R7, R8, R13, R34	0805, res0805	4	100K	
Resistor	R9	0805	1	12K	
Resistor	R10	0805	1	47K	
Resistor	R12	0805	1	140	
Header, 3-Pin	R Power, RS1, RS2, RS3, RS4, RS5, RS6, RS7, S1, S2, S3, S4, S5, S6, S7, SEL	HDR1X3	16		
Microchip PIC18F458	U1	DIP40	1		
QUAD 2 input NAND,74HC132	U2	DIP14	1		
Dual Operational Amplifier	U3	P008	1		
Differential Comparator with Strobes	U4	P008	1		
5V powered RS232 Driver/Tranceiver	U5	SOIC16L	1		3530036
	U6, U7	DIP16	2		
Crystal	X1	HC49	1	40 MHz	


U_Gyrointerface
Gyrointerface.SCHDOC
V5.0
JP1
5VPC104
JP2
5V Power pc104
JP3
U_Superstar2Interface
Superstar2Interface.SCHDOC
V5.0
V3.3
U_powerSupply
powerSupply.SCHDOC
V3.3
V5.0
Title
PC104 extension card
Size
A4
Number
Revision
Date:
01-06-2005
Sheet of
File:
D:\miss\...\pc104extensionCard.SCHDOC
Drawn By:
Group 1034, 2005


Title
Switch Mode Power Supplies, part of PCI 04 extension card
Size
A4
Number
Revision
Date: 01-06-2005
Sheet of
File: D:\mischardware\powerSupply.SCHDOC
Drawn By: Group 1034, 2005
JP4
JP5
JP6
JP7
V5.0
V3.3
VBat
L1
33uH
L2
68uH
D1
40V.3A
D2
40V.1A
C1
10nf
C2
470pf
C3
100nf
C4
10nf
C5
15uf/20V
C6
100uf/10V
C7
15uf/20V
C8
100uf/10V
C9
100uf/16V
C10
100uf/10V
R1
10K
U1
LM2675-5.0
U2
LM2675-3.3
Vin
GND
Vsw
CB
FB
SS
ON/OFF


Title
Gyro interface, part of the PC104 extension card
Size
A4
Number
Revision
Date: 01-06-2005
File: D:\mies\Hardware\GyroInterface.SCHDOC
Sheet of
Drawn By: Group 1034, 2005
R3
10K
JP12
Header 9X2
AIN8
AIN16
AIN15
AIN14
AIN13
AIN12
AIN11
AIN10
AIN9
P1.[0..7]
JP11
AIN1..16]
AIN1
AIN2
AIN3
AIN4
AIN5
AIN6
AIN7
AIN8
AIN9
AIN10
AIN11
AIN12
AIN13
AIN14
AIN15
AIN16
EGATE0
AGND
P1.7
P1.6
P1.5
P1.4
P1.3
P1.2
P1.1
P1.0
P0.7
P0.6
P0.5
P0.4
P0.3
P0.2
P0.1
P0.0
TCOUT0
TCOUT1
TCOUT2
EGATE2
+5V
GND
ECLK0
EGATE1
ECLK1
ECLK2
+12V
-12V
compPC104IO
P0.[0..7]
JP13
JP14
ST12
V5.0


Title
GPS Interface, part of the PC104 extension card
Size
A4
Number
Revision
Date: 01-06-2005
Sheet of
File: D:\miso\...\Superstar2Interface.SCHDOC
Drawn By: Group 1034, 2005
JP8
Header 10X2H
JP10
RS232Terminal
JP9
GPS Mreser
U3
ST232
R2
10K
C11
C12
C13
C14
C15
100nf
V5.0
V3.3
RXD_NO_3
DISC_IP1
DISC_IP3
GND
RXD_NO_1
TXD_NO_2
GND
GND
TXD_NO_3
VBATT
MASTER
DISC_IP2
DISC_IO1
TXD_NO_1
GND
RXD_NO_2
DISC_IO2
TIMEMARK
DSRRS232
CTSRS232
RTSRS232
RIRS232
GND
DCDRS232
TXRS232
RXRS232
DTRRS232
GND
V+
V-
VCC
GND
C1+
C1-
C2+
C2-
T1i
T2i
R1o
R2o
TX1
TX2
RX1
RX2

PC104 Extension Board: Bill Of Material Group 1034, 2005

Description	Designator	Footprint	Quantity	Value	Farnell code
ceramic capacitor	C1, C4	0805	10	10nf	
ceramic capacitor	C2	0805	2	470nf	
ceramic capacitor	C3, C11, C12, C13, C14, C15	0805	1	100nf	
Tantalum	C5, C7	case B	1	15uf/20V	197439
Tantalum	C6, C8, C10	case D	1	100uf/10V	757780
Tantalum	C9	case D	5	100uf/16V	197348
Schottky Diode	D1	SMC	1	40V,3A	506758
Schottky Diode	D2	SMA 403D-02	1	40V,1A	3004340
PC104 A-B Dual row 2x32pin interface connector	JP1	PC104AB	1		
Header, 2-Pin, Dual row (jumper)	JP2	HDR2X2	2		
PC104 C-D Dual row 2x20pin interface connector	JP3	PC104CD	1		
Robust vertical 2 pin connector (Orange)	JP4, JP5, JP6	conVert2pinRobust	10		
Robust vertical 3 pin connector (Orange)	JP7	conVert3pinRobust	14		
Header, 2x10-Pin, Dual row, Right Angle	JP8	HDR2X10H	1		
Header, 2-Pin (jumper)	JP9	HDR1X2	1		
Header, 2x5-Pin, Dual row	JP10	HDR2X5	1		
Dual row 2x25 pin connector	JP11	con2x25HA	4		
Header, 2x9-Pin, Dual row (kumper)	JP12	HDR2X9	1		
JST ZH 12pin connector (1x12pin) 1.5mm pitch	JP13	ZHCon	1		Digikey
Header, 2x3-Pin, Dual row (jumper)	JP14	HDR2X3	1		
Inductor	L1	b82477G4	16	33uH	7430396
Inductor	L2	SMID 2202-A	1	68uH	158781
Resistor	R1, R2, R3	0805	1	10K	
Simple switcher 3A Step-Down Voltage Regulator	U1	TO-263	1		3000576
Simple switcher 1A Step-Down Voltage Regulator	U2	SO-G8	1		4125575
5V powered RS232 Driver/Tranceiver	U3	SO-G16/D10.4	1		3530036

Index

$C_{\mathbb{B}\boldsymbol{F}_A}$, 22
$C_{\mathbb{B}\boldsymbol{\tau}_A}$, 22
$C_{\mathbb{A}F_{x,input}}(\alpha, \boldsymbol{\delta})$, 27
$C_{\mathbb{A}F_{x,zero}}$, 27
$C_{\mathbb{A}F_{y,input}}(\boldsymbol{\delta})$, 29
$C_{\mathbb{A}F_{y,zero}}(\alpha, \beta, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}, V_\infty)$, 28
$C_{\mathbb{A}F_{z,0}}$, 32
$C_{\mathbb{A}F_{z,input}}(\boldsymbol{\delta})$, 32
$C_{\mathbb{A}F_{z,zero}}(\alpha, \dot{\alpha}, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}}, V_\infty)$, 30
$C_{\mathbb{B}\tau_x,input}(\boldsymbol{\delta})$, 35
$C_{\mathbb{B}\tau_x,zero}$, 34
$C_{\mathbb{B}\tau_y,0}$, 38
$C_{\mathbb{B}\tau_y,input}(\alpha, \boldsymbol{\delta})$, 38
$C_{\mathbb{B}\tau_y,zero}(\alpha, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}})$, 38
$C_{\mathbb{B}\tau_z,input}(\alpha, \boldsymbol{\delta})$, 41
$C_{\mathbb{B}\tau_z,zero}(\alpha, \beta, {}^{\mathbb{B}}\boldsymbol{\omega}_{\mathbb{B}})$, 41
S, 22
V_∞, 18
α, 18
$\bar{q}$, 22
β, 18
$\mathbb{A}$, 15
$\mathbb{B}$, 13
$\mathbb{E}$, 13
$\mathbb{N}$, 15
$\mathbb{R}$, 15
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{x,input}}(\alpha)$, 27
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{x,zero}}(\alpha)$, 27
$\boldsymbol{P}_{\mathbb{A}F_{y,input}}$, 29
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{y,zero}}(\alpha)$, 29
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{z,input}}(\delta_f)$, 32
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{A}F_{z,zero}}(\alpha)$, 32
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{x,input}}$, 35
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{x,zero}}(\alpha)$, 34
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{y,input}}(\alpha, \boldsymbol{\delta})$, 38
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{y,zero}}(\alpha)$, 38
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{z,input}}(\alpha)$, 41
$\boldsymbol{P}^{\mathrm{T}}_{\mathbb{B}\tau_{z,zero}}(\alpha)$, 41
b, 22
c, 22

Taylor linearization, 55

Aerodynamic Coefficient
    Drag, 25
    Lift, 25, 30
    Pitch Torque, 25, 35
    Roll Torque, 25, 33
    Side Force, 25, 28
    Yaw Torque, 25
Aerodynamic Coefficients, 24
Aerodynamics, 21
    Drag, 16
    Lift, 16
    Skin Friction, 16, 27
    Torque, 17
Aileron, 12
Aircraft
    Aileron, 17
    control surfaces, 17
    Elevator, 17
    Fuselage, 13, 24
    High Winged, 69
    Rudder, 17
Aircraft centerline, 13
Angle of attack, 18

Boundary Layers, 25

Coefficient
    Yaw Torque, 39
Controller
    Performance, 63

Coordinate Systems
    Aerodynamic, 24
Coordinate systems, 13
    Aerodynamic, 15
    Body-fixed, 13
    Control Reference, 15
    Earth-fixed, 13
    Navigational, 15

Definitions, 12
dimension less, 22
Drag, 16
dynamic pressure, 22

Elevator, 13
EQuaternion
    uler's eigenaxis, 95

Feedback linearization, 55
Flaps, 13
Fuselage, 13, 24

High Winged Aircraft, 69
Horizontal Stabilizer, 13

Induced Drag, 24
Induced Flow, 24
Inertial system, 13
Interconnected System, 66, 67
Interconnected Systems, 66
Internal Dynamic, 58
Internal Dynamics, 66, 67

Lie Derivative, 56
Lift, 16
Lyapunov
    Asymptotically Stable, 66
    Direct Method, 66
    Interconnected Systems Approach, 66
    Second Method, 66

MIMO system, 57

Noninteractive, 58
Nonlinear
    Normal Form, 67
Normal Form, 67

Propeller, 13

Quaternion
    ⊗, 96
    conjugated, 95
    control law, 61
    controller, 59
    Euler parameters, 95
    example, 96
    fundamental formula, 95
    hypercomplex number, 95
    multiplication, 96
    norm, 96
    skew-symmetric matrix, 96
    time derivative, 97, 98

Relative Degree
    SISO, 55
Relative degree, 56
relative wind speed, 18
Rudder, 13

Side Slip, 18
Skin Friction, 27
Stability, 66
state-space controller, 59

The Aerodynamic Model, 48

Vertical Stabiliser, 13
Vertical Stabilizer, 24
VLM, 22, 24
Vortex Lattice Method, 22

Wind Tunnel Data, 24
Wing, 12